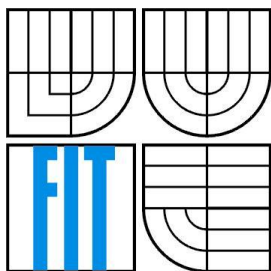


**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

# **NÁVRH A IMPLEMENTACE KOMPONENTY PRO KOMUNIKACI S PCI ROZHRANÍM**

DESIGN AND IMPLEMENTATION OF THE COMPONENT FOR COMMUNICATION WITH PCI  
INTERFACE

## **DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. MICHAL JANOUŠEK**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. TOMÁŠ MARTÍNEK, PhD.**

BRNO 2011

## Zadání diplomové práce

Řešitel: **Janoušek Michal, Bc.**

Obor: Počítačové a vestavěné systémy

Téma: **Návrh a implementace komponenty pro komunikaci s PCI rozhraním**  
**Design and Implementation of the Component for Communication with PCI Interface**

Kategorie: Počítačová architektura

Pokyny:

1. Seznamte se s kartou COMBO6-PTM a technologií Spartan 3 od firmy Xilinx.
2. Nastudujte specifikaci obvodu PLX, který je součástí karty COMBO6-PTM a způsobem komunikace přes sběrnici PCI.
3. Na základě poznatků z bodu 1 a 2 navrhnete vhodnou architekturu komponenty pro komunikaci s PCI prostřednictvím obvodu PLX.
4. Proveďte implementaci navržené komponenty v jazyce VHDL a ověřte její správnost simulací.
5. Realizujte prototyp na kartě COMBO6-PTM a ověřte její funkčnost.
6. Zhodnoťte dosažené výsledky a diskutujte možné pokračování projektu.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části diplomového projektu je požadováno:

- Splnění bodů 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Martínek Tomáš, Ing., UPSY FIT VUT**

Datum zadání: 20. září 2010

Datum odevzdání: 25. května 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
Fakulta informačních technologií  
Ústav počítačových systémů a sítí  
602 00 Brno, Božetěchova 2



doc. Ing. Zdeněk Kotásek, CSc.  
vedoucí ústavu

## **Abstrakt**

Tato diplomová práce se zabývá návrhem komponenty zprostředkovávající komunikaci s PCI sběrnici dalším uživatelským komponentám. Návrh je veden tak, aby došlo k výraznému zjednodušení komunikačního protokolu mezi navrženou a uživatelskou komponentou a pokročilé funkce sběrnice PCI zůstaly zachovány. Jako cílová platforma byla použita karta COMBO6-PTM obsahující FPGA s technologií Spartan 3. Komunikace se sběrnici PCI probíhá prostřednictvím obvodu PLX. Součástí práce je také návrh zjednodušeného komunikačního protokolu.

## **Abstract**

This master's thesis deals with design of the component facilitating communication between PCI bus and user component. Designed component is simplifying the communication protocol between designed and user components, while advanced functions of PCI bus are preserved. Target platform is COMBO6-PTM card containing FPGA with Spartan 3 technology. Communication with PCI bus is mediated by PLX component. Thesis also contains design of simplified communication protocol.

## **Klíčová slova**

PCI, DMA, FPGA, Spartan 3, PLX, VHDL, komunikační protokol

## **Keywords**

PCI, DMA, FPGA, Spartan 3, PLX, VHDL, communication protocol

## **Citace**

Janoušek Michal: Návrh a implementace komponenty pro komunikaci s PCI rozhraním, diplomová práce, Brno, FIT VUT v Brně, 2011

# Návrh a implementace komponenty pro komunikaci s PCI rozhraním

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Tomáše Martínka, PhD. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Bc. Michal Janoušek

24.5.2011

## Poděkování

Děkuji vedoucímu projektu Ing. Tomáši Martínkovi, PhD., za velice odborné vedení a cenné připomínky.

© Bc. Michal Janoušek, 2011

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů*

# OBSAH

<b>1</b>	<b>Úvod .....</b>	<b>2</b>
1.1	Motivace.....	2
1.2	Cíl práce .....	2
1.3	Struktura práce.....	3
<b>2</b>	<b>Teoretický rozbor .....</b>	<b>4</b>
2.1	Architektura systému s akcelerační kartou .....	4
2.2	Karta Combo6-PTM .....	5
2.3	Sběrnice PCI .....	6
2.4	Přímý přístup do paměti (DMA).....	10
2.5	Vestavěné jádro PLX typ PCI 9054 .....	11
<b>3</b>	<b>Návrh.....</b>	<b>17</b>
3.1	Komunikační protokol uvnitř FPGA.....	18
3.2	Časová souslednost užití protokolů v rámci operací .....	20
3.3	Architektura základní komponenty .....	23
3.4	Architektura rozšířené komponenty.....	27
<b>4</b>	<b>Simulace a syntéza .....</b>	<b>39</b>
4.1	Testovací prostředí .....	39
4.2	Základní komponenta.....	39
4.3	Rozšířená komponenta.....	42
<b>5</b>	<b>Závěr .....</b>	<b>45</b>
<b>A</b>	<b>Obsah přiloženého CD .....</b>	<b>47</b>
<b>B</b>	<b>Schéma základní komponenty .....</b>	<b>48</b>
<b>C</b>	<b>Schéma konečného automatu základní komponenty .....</b>	<b>49</b>
<b>D</b>	<b>Časové diagramy operací základní komponenty .....</b>	<b>50</b>
<b>E</b>	<b>Schéma rozšířené komponenty .....</b>	<b>51</b>
<b>F</b>	<b>Schéma konečného automatu rozšířené komponenty .....</b>	<b>52</b>
<b>G</b>	<b>Časové diagramy operací rozšířené komponenty .....</b>	<b>53</b>

# 1 Úvod

## 1.1 Motivace

V současné době, která je charakteristická především rostoucím požadavkem na výpočetní výkon počítačových systémů, se do popředí jako jedno z možných řešení k uspokojení těchto požadavků dostávají hardwarová zařízení vytvořená k urychlení časově kritických částí zpracovávaných úkolů. Praxe ukazuje, že tato specializovaná zařízení v oblastech svého zaměření výkonnostně převyšují „klasické“ univerzální výpočetní systémy jako jsou například systémy s univerzálními mikroprocesory.

K tvorbě těchto specializovaných systémů je využíváno několika přístupů a metod. Tyto se mezi sebou liší délkou vývoje, technologickou náročností výroby a v neposlední řadě také výrobní cenou. V poslední době se začínají stále častěji používat jako jedno z cenově dostupných řešení programovatelná logická zařízení tzv. PLD, či jejich výkonnější nástupce FPGA - programovatelná hradlová pole. Jedná se o velice flexibilní logická zařízení, jejichž konkrétní funkce je tvořena teprve koncovým uživatelem a tuto lze v průběhu užívání opakovaně měnit. Díky této vlastnosti našla programovatelná hradlová pole širokou škálu užití, jmenujme například analyzátoři síťových toků [1], systémy pro podporu výpočtů DNA[2], medicínské systémy[3] apod.

Jedno z možných využití technologie FPGA je kombinace konvenčního počítačového systému např. běžné stolní PC s přídatnou akcelerační kartou osazenou jedním či více FPGA. Konvenční nebo přesněji univerzální část tohoto systému je využívána pro zpracování obslužné části aplikací a výkonnostně náročné výpočty jsou svěřovány účelově naprogramovanému čipu FPGA. Jedním z problémů těchto systémů je komunikace mezi jeho univerzální částí (obecný mikroprocesor, paměť RAM) a akcelerační kartou. Tato je většinou realizována pomocí univerzální sběrnice, jejíž komunikační protokol je pro účely výměny informací s akcelerační kartou příliš komplexní. Z tohoto důvodu jsou do akceleračních karet umísťována tzv. vestavěná jádra pro komunikaci s univerzální sběrnicí (někdy IP cores), která komunikační protokol částečně zjednodušují. I přes to však zůstala potřeba vytvořit zařízení, které v maximální míře usnadní komunikaci s konvenční částí počítačového systému a přitom zachová pokročilé vlastnosti univerzální sběrnice. Užitím takového zařízení jako zprostředkovatele přístupu k PCI pak dojde zejména k výraznému urychlení vývoje a snížení komplexnosti dalších aplikací určených pro dané FPGA, přičemž zejména u hardwarových zařízení vede toto snížení komplexnosti aplikací ke zvýšení maximální dosažitelné rychlosti (zkrácení kritických cest - zařízení jsou schopna pracovat na vyšším kmitočtu) a k omezení počtu potřebných hardwarových prostředků programovatelného hradlového pole.

## 1.2 Cíl práce

Cílem diplomové práce je vytvoření komponenty, které na straně aplikací umístěných v FPGA výrazně zjednoduší komunikační protokol, přičemž dojde k zachování pokročilých

funkcí sběrnice, jako jsou dávkové (BURST) a DMA (přímý přístup do paměti) přenosy. Konkrétní platformou projektu je hardwarová karta COMBO6-PTM osazená programovatelným hradlovým polem Spartan 3. Ke konvenční části běžného stolního počítače je karta připojena sběrnici PCI skrze vestavěné jádro pro komunikaci se sběrnici čipem PLX. Komponenta musí být navržena tak, aby mohla být provedena implementace a následná syntéza do programovatelného hradlového pole Spartan 3. Součástí projektu je také návrh jednoduchého komunikačního protokolu uvnitř tohoto FPGA.

### 1.3 Struktura práce

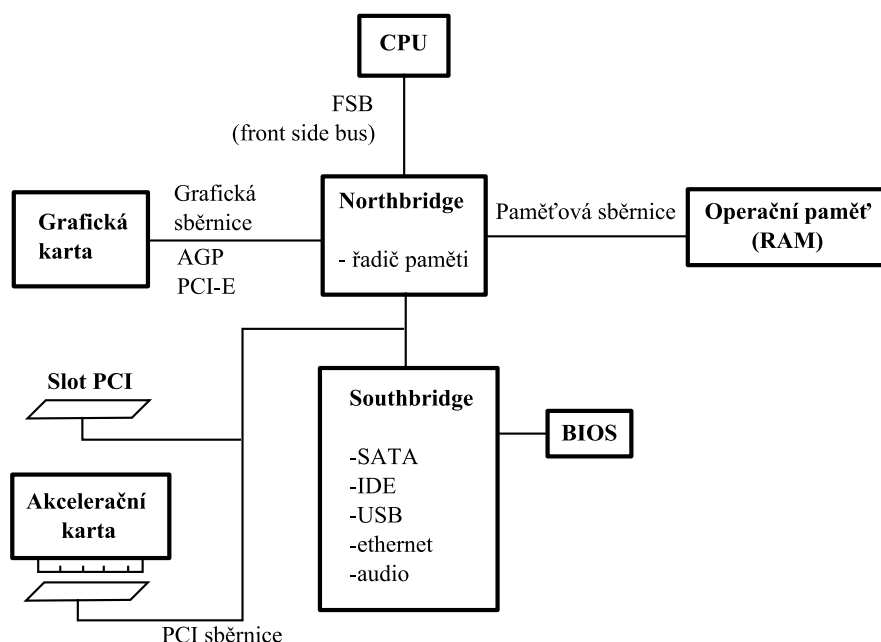
Dokument je rozvržen do pěti kapitol. Po úvodu následuje teoretický rozbor zadání, v něm je rozebrána architektura celého systému a technické specifikace jednotlivých použitých zařízení. Součástí rozboru je struktura samotné karty COMBO6-PTM, dále struktura a komunikační protokol sběrnice PCI včetně specifikace zařízení PLX, které je umístěno mezi FPGA Spartan 3 a patičí sběrnice PCI.

Poté následuje kapitola 3 - Návrh, kde je popsány vytvořené komunikační protokoly užívané uvnitř FPGA Spartan 3. Je zde také podrobně popsáno navržené zařízení v jeho základní i rozšířené formě. Čtvrtou kapitolou jsou pak provedená testování a simulace navržených komponent. Poslední kapitolou č. 5 je závěrečné shrnutí práce včetně zhodnocení dosažených výsledků a určení dalšího možného vývoje navržených zařízení.

## 2 Teoretický rozbor

### 2.1 Architektura systému s akcelerační kartou

Programovatelná hradlová pole mají široké spektrum užití např. jako součást vestavěných systémů, inteligentních zařízení apod. Jedním z cenově a technologicky přístupných způsobů je jejich použití jako součásti tzv. akceleračních karet, tedy karet, které jsou určeny pro podporu široké škály pomocných výpočtů a výkon nejrozličnějších specializovaných funkcí. Jejich velkou výhodou je možnost na tyto karty umístit libovolné další komponenty, a tak značně rozšířit množinu jejich užití. Karty jsou takto doplňovány o operační paměti různých velikostí, výkonná síťová rozhraní, jednotky pro měření fyzikálních veličin, inteligentní čidla apod.



Obrázek 2.1: Příklad architektury osobního PC s akcelerační kartou

Společnou vlastností akceleračních karet je, že tyto jsou umísťovány jako součást dostupných konvenčních výpočetních systémů. Těmito systémy jsou ve většině případů univerzální osobní počítače. Existuje mnoho způsobů, jak akcelerační karty do těchto systémů začlenit. Jedním z nich je zapojení akcelerační karty přímo do slotu univerzální sběrnice. Tento způsob je velice výhodný nejen z pohledu rychlé odezvy takto připojeného zařízení, ale vykazuje také velice příznivou přenosovou rychlost. Nevýhodou pak je větší složitost komunikačního protokolu univerzální. Další alternativou je připojení karty skrze pomalejší lokální sběrnice jako je již starší paralelní či dnes masově užívaný sériový port např. USB



(Universal Serial Bus). Nevýhodou tohoto způsobu zapojení je nízká propustnost a nutnost připojit k zařízení také zdroj napájení (vyjma USB u energeticky nenáročných zařízení). V rámci práce je významný především první způsob připojení, neboť užitá akcelerační karta COMBO6-PTM se připojuje přímo do slotu sběrnice PCI.

Architektura systému užitého v rámci projektu, tedy běžného stolního PC a akcelerační karty připojené skrze sběrnici PCI, je zobrazena na obrázku 2.1. Obsluha takto připojené akcelerační karty probíhá skrze aplikační rozhraní běžící na univerzální části sestaveného výpočetního systému.

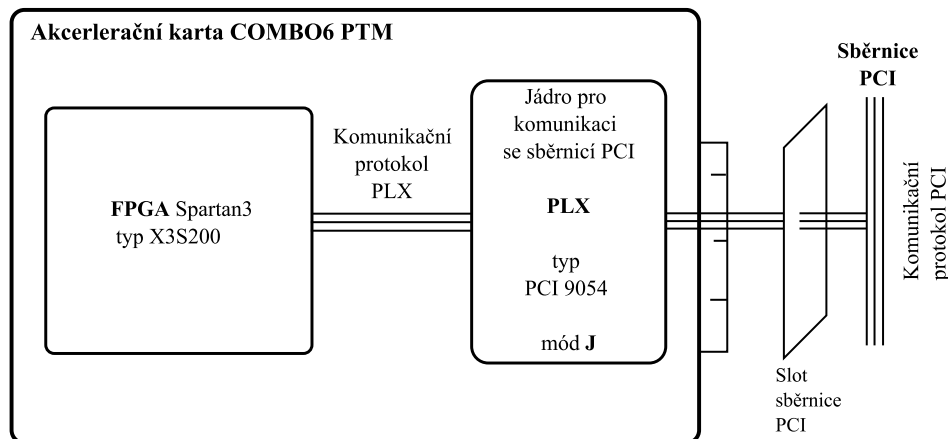
## 2.2 Karta Combo6-PTM

Akcelerační karta COMBO6-PTM[5] byla vyvinuta pro potřeby projektu Liberouter[1] zabývajícího se vývojem hardwarově akcelerovaných aplikací pro vysokorychlostní síť. Jedná se o jednu z řady akceleračních užívaných v rámci tohoto projektu. Karta byla původně určena jako zdroj přesných časových známek, ale lze ji úspěšně využít také jako univerzální akcelerační kartu. Díky své univerzalitě je používána i v laboratoři CESNET pro účely výuky.



Obrázek 2.2: Akcelerační karta Combo6-PTM  
(převzato z [5])

Karta COMBO6-PTM (obrázek 2.2) je osazena programovatelným hradlovým polem Spartan3 typ X3S200, procesorem Texas Instruments typ MSP430FI49IPM, vestavěným jádrem pro komunikaci s univerzální sběrnici PLX typ PCI9054, programovatelným logickým zařízením CPLD typ XCR3256XL, pamětí EEPROM 93S66 a sériovým portem RS232/485.



Obrázek 2.3. Schéma zapojení Combo6-PTM do PCI

Pro účely této práce je důležitý pouze podsystém karty COMBO6-PTM sestávající z jádra pro komunikaci se sběrnici PCI (PLX), programovatelného hradlové pole Spartan3 a zásuvného portu do slotu PCI. Zjednodušené schéma zapojení včetně zobrazení jednotlivých komunikačních protokolů těchto hardwarových prvků je na obrázku 2.3.

## 2.3 Sběrnice PCI

PCI (z anglického Peripheral Component Interconnect) je univerzální počítačová sběrnice určená pro připojení periférií k základní desce. Její využití však není omezeno jen na platformu osobních počítačů. Používá synchronní paralelní přenos dat se šířkou 32 nebo 64 bitů a je orientovaná na přenos zpráv (anglicky message passing). Od zbytku systému je oddělena pomocí PCI mostů, které zprostředkovávají komunikaci s připojenými kartami. V jednom počítači může být i více nezávislých PCI sběrnic. Jejím prostřednictvím lze připojit periférie ve formě integrovaného obvodu zabudovaného přímo do základní desky někdy označovaného jako planární zařízení nebo jako tzv. rozšiřující kartu, která se vsune do konektoru (slotu) sběrnice. Sběrnice PCI je běžnou součástí osobních počítačů, kde jako standardní rozšiřující sběrnice nahradila sběrnici ISA a VESA Local Bus. [7] Poslední verzi sběrnice PCI je verze 3.0. V současnosti je definována jako standard.

Sběrnice PCI je charakteristická několika specifickými vlastnostmi. První z nich je možnost automatické konfigurace připojených zařízení PnP (z anglického Plug and Play), dále využívá systém iniciování přenosů na sběrnici ze strany připojených zařízení (anglicky Busmastering), čímž dochází k výraznému snížení zatížení procesoru, a v neposlední také řadě umožňuje sdílení jednotlivých přerušení. U současné verze PCI se systém sdílení přerušení již nepoužívá a je nahrazen systémem signalizace přerušení pomocí zasilání zpráv.

**Komunikační protokol** sběrnice PCI přesně specifikuje časový sled událostí, které nastanou v průběhu jednotlivých přenosových a konfiguračních transakcí. Úvod této části

kapitoly se věnuje komunikačnímu protokolu a je v něm shrnut standardní průběh přenosu se zaměřením na činnosti jednotlivých účastníků sběrnice transakce, ke konci pak budou ukázány dva základní způsoby komunikace včetně příslušných časových diagramů. Z důvodu snížení počtu vodičů je k přenosu adres i dat užito sdíleného kanálu sestávajícího z 32 nebo 64 vodičů (signál AD). Veškeré transakce na sběrnici pracují synchronně k hodinovému signálu (CLK) sběrnice.

Zařízení, které se rozhodne vysílat (anglicky initiator), musí nejprve získat povolení ke kontrole nad sběrnici. Fyzicky je tento proces realizován tak, že příslušná část čipové sady sběrnice PCI na základě vložených algoritmů rozhodne o přidělení práva ke kontrole provozu na sběrnici jednomu z žádajících zařízení. Tento proces zvaný arbitráž je specificky navržen tak, aby se předcházelo nežádoucím jevům, jako jsou např. vyhladovění, či monopolizace sběrnice jedním z připojených zařízení.

Jakmile je proces výběru dokončen, jeho vítěz na sběrnici vystaví cílovou adresu (signál AD), kódové označení reprezentující typ přenosu (signál C/BE[4:0]) a nastaví signál FRAME. Základní druhy typů přenosů jsou uvedeny v tabulce 2.1. Adresované zařízení potvrdí zahájení přenosu vystavením signálu DEVSEL (celým názvem DEVice SElect). Jakmile adresované zařízení potvrdí zahájení, celý přenos vstoupí do své datové fáze, která může sestávat z jednoho či více hodinových taktů.

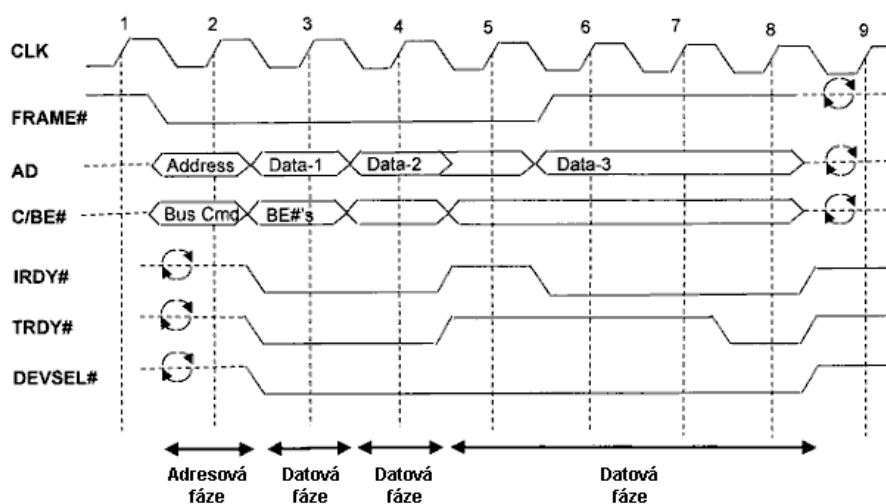
C/BE[3:0]	Typ přenosu
0	Potvrzení přerušení
1	Speciální cyklus
10	I/O čtení
11	I/O zápis
110	Čtení z paměti
111	Zápis do paměti
1010	Čtení konfigurace
1011	Zápis konfigurace
1100	Vícenásobné čtení z paměti
1101	Cyklus s rozšířenou adresou

Tabulka 2.1 Typy přenosů na sběrnici PCI

Během každého taktu datové fáze je skrze signál AD přeneseno množství dat odpovídající šířce sběrnice, tedy 32 nebo 64 bitů. Datový přenos sestávající z přenosu několika datových bloků v rámci více hodinových taktů se nazývá dávka (anglicky BURST). Datový přenos může být ukončen několika způsoby. Prvním je přenesení veškerých požadovaných dat, což je indikováno shobením signálu FRAME. Mezi další způsoby pak patří ukončení ze strany vysílajícího nebo cílového zařízení, která nejsou schopna v delším čase odeslat či přijmout další

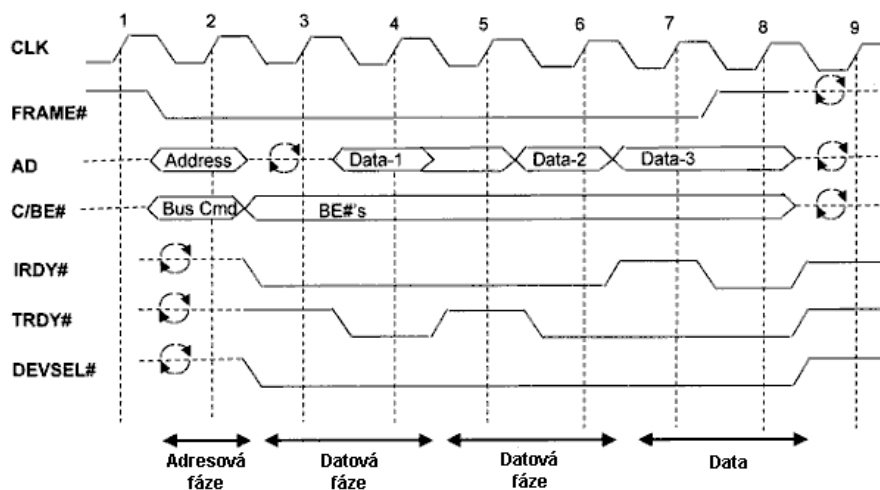
část přenášených dat. V případě, že dojde k této situaci ještě před započítím datové části transakce, využije se příkazu RETRY. U započatých transakcí se využívá příkazu DISCONNECT. Jeho vložení indikováno, že se má transakce zahájit znovu ale později, kdy by již odeslání nebo příjem dalších dat měl být připraven. Poslední možností ukončení přenosu je přerušení ze strany zařízení provádějícího arbitráž, které takto předchází nežádoucím situacím jmenovaným výše.

V případě přerušení transakce nelze již v započaté transakci pokračovat a je nutno zahájit celý přenosový cyklus znovu jako novou transakci.[9] Z důvodu vyšší efektivity je žádoucí, aby datové přenosy probíhaly v dávkovém režimu (BURST).



Obrázek 2.4 Průběh zápisové operace na sběrnici PCI (převzato z [9])

Zařízení, která nejsou krátkodobě schopna vyslat či přijmout další data mají možnost, jak se přerušení transakce vyhnout. Pro tyto účely mohou do přenosové transakce vložit až 16 tzv. čekacích taktů. Tyto mohou být iniciovány jak ze strany vysílajícího zařízení (signálem IRDY), tak i ze strany cílového zařízení (signál TRDY). Specifikace dalších operací realizovaných na sběrnici lze najít v příslušné normě standardu PCI v 3.0. Na obrázcích 2.4 a 2.5 je zobrazen průběh základní čtecí a zápisové operace v dávkovém režimu.



Obrázek 2.5 Průběh čtecí operace na sběrnici PCI  
(převzato z [9])

Konfliktní situace, kdy je na sběrnici připojeno více zařízení se schopností iniciovat přenos a některá z těchto zařízení uplatní své požadavky ve stejném čase, je na sběrnici PCI řešena pomocí metody zvané **arbitráž**. Zařízení zastávající tuto funkci se nazývá **arbitr**. Jeho úkolem je řešení všech možných situací, které mohou nastat mezi skupinou připojených zařízení. V současné době bývá arbitr sběrnice PCI umísťován jako součást komponenty Northbridge. (viz. obrázek 2.1)

Hlavním cílem arbitráže sběrnice je zajistit spravedlivý přístup všech zařízení ke sběrnici. Pro účely úspěšné arbitráže mají všechna zařízení schopná iniciovat přenos v sobě obsažen tzv. konfigurační registr, ve kterém je uložen údaj o maximální době, po kterou tato zařízení mohou vysílat.

Z důvodu snížení latence sběrnice je proces arbitráže prováděn na pozadí (skrytě). To znamená, že je arbitráž zahájena již v době, kdy na sběrnici stále probíhá jiná transakce a ještě před dokončením této transakce je vybrán její vítěz. Takto vybrané zařízení pak může zahájit další vysílání bezprostředně po ukončení běžící transakce. Žádost o účast v arbitráži je signalizována signálem REQ a vítězství pak potvrzeno signálem GNT. Zařízení, které takto obdrží signál GNT, proto sleduje provoz na sběrnici a přenos započne, až jakmile se sběrnice uvolní.

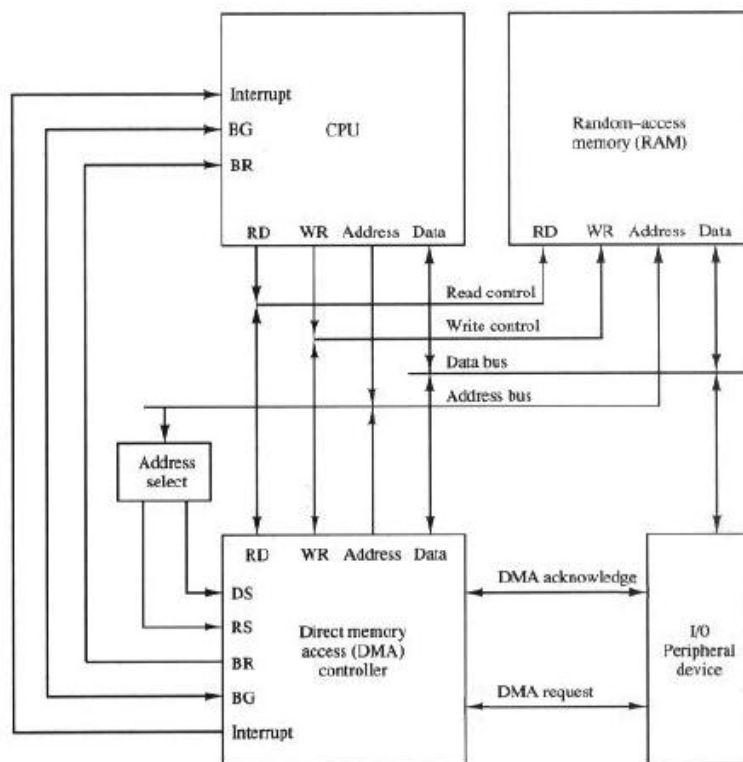
Ne vždy je však vhodné, aby všechna zařízení měla rovnocenná práva k přístupu ke sběrnici. Z tohoto důvodu jsou do výpočtu vítěze arbitráže zahrnuty také tzv. prioritní úrovně. Jedním z hlavních parametrů pro výpočet prioritní úrovně je maximální doba vysílání uložená v konfiguračním registru.

V případě, že o právo vysílat požádá zařízení, které má vyšší úroveň priority než zařízení, kterému již právo vysílat bylo signálem GNT přiděleno, a tato transakce ještě nezačala, arbitr může již přidělené právo odebrat a přidělit ho zařízení s vyšší prioritou. Pokud však již transakce započala, arbitr k rozhodnutí použije jiný mechanismus. Každé zařízení má v sobě

kromě konfiguračního registru vložen také registr s minimální dobou vysílání. Tento údaj se v každém cyklu, kdy zařízení provádí transakci, snižuje. K odebrání práva vysílat ze strany arbitra po žádosti od zařízení s vyšší prioritou pak může dojít pouze v případě, že aktuálně vysílající zařízení již tuto minimální dobu překročilo a údaj v jeho registru s minimální dobou je roven nule.

## 2.4 Přímý přístup do paměti (DMA)

DMA (Direct Memory Access) je způsob, kterým počítače umožňují hardwarovému subsystému přímý přístup do operační paměti tzn. přístup bez účasti CPU. Pomocí DMA mohou počítače přenášet data mezi zařízeními a operační pamětí s podstatně menšími nároky na procesor než by tomu bylo u zařízení bez DMA. [10] U starších typů sběrnic byly DMA přenosy realizovány a řízeny pomocí specializovaného řadiče (obrázek 2.6).



Obr 2.6 Systém přímého přístupu do paměti s DMA řadičem  
(převzato z [10])

U sběrnice PCI jsou však veškeré transakce realizovány odlišným způsobem zvaným „bus mastering“. Veškerá zařízení připojitelná do sběrnice PCI schopná iniciovat přenosovou

transakci se pak nazývají „bus masters“. To znamená, že jsou schopna převzít kontrolu nad sběrnici a veškeré přenosy provádět v rámci vlastní režie. Přímé přístupy do paměti (DMA) jsou pak jen určitou podmnožinou takovýchto obecných přenosů. V případě DMA zápisů je cílovým zařízením transakce komponenta „Northbridge“, která pak zápis do operační paměti fyzicky realizuje.

## 2.5 Vestavěné jádro PLX typ PCI 9054

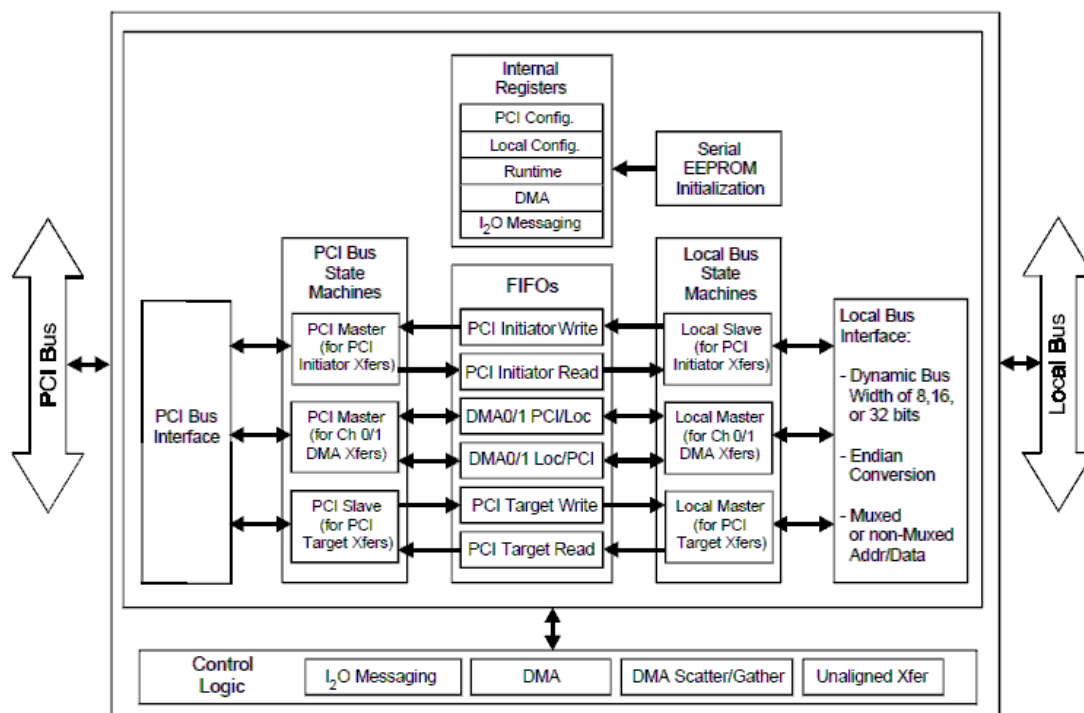
**Vestavěné jádro PLX** je komplexním zařízením určeným pro univerzální použití v rámci vestavěných číslicových systému nebo přídavných výpočetních karet do sběrnice PCI. Plní funkci rozhraní mezi lokální sběrnici reprezentovanou hradlovým polem a sběrnici PCI. Jeho cílem je zjednodušit a zpřístupnit komplexní komunikační protokol sběrnice PCI a současně poskytnout hardwarové prostředky potřebné k plnému využití potenciálu této sběrnice. V této kapitole je popsána specifikace použitého zařízení PLX se zaměřením na funkce využívané v rámci této práce. V úvodu jsou shrnuty obecné vlastnosti zařízení a nastíněna jeho struktura, následuje popis použitého komunikačního módu zařízení a na závěr je pak podrobně rozebrán komunikační protokol, který pracuje na rozhraní s lokální sběrnici. Tento protokol je použit právě pro komunikaci s programovatelným hradlovým polem Spartan 3 použitým na kartě COMBO6-PTM v rámci projektu.

Zařízení pracuje s PCI sběrnici verze 2.2 s 32 bitovou datovou šířkou při kmitočtu 33 MHz. Mezi jeho přednosti patří rozšířená podpora dávkových přenosů (Burst), zařízení v sobě obsahuje dvoukanálový řadič DMA včetně módů scatter/gather a má podporu pro práci v režimech PCI Initiator (PCI master) a PCI Target (PCI slave) s možností tzv. dvou adresového cyklu (rozšiřuje tím rozsah adresovatelných zařízení na 64bitů). [12]

Režim	Popis
M	32-bitů adresa, 32-bitů data, nesdíleno, určeno pro procesory Motorola
C	32-bitů adresa, 32-bitů data, nesdíleno
J	<b>32-bitů adresa, 32-bitů data, sdíleno</b>

Tabulka. 2.2 Režimy zařízení PLX typ PCI 9054

Zařízení je schopno pracovat ve třech režimech (tabulka. 2.2), v rámci projektu se využívá pouze režimu „J“. Vybraný režim ovlivňuje zapojení a komunikační protokol na straně rozhraní s lokální sběrnici. Zvolenému režimu odpovídá fyzické zapojení jednotlivých pinů čipu PLX, režim je tedy pevně zvolen a nelze jej v průběhu užívání měnit.



Obrázek 2.8 Interní blokové schéma čipu PLX PCI 9054  
(převzato z [12])

PLX pracuje na každém z rozhraní (PCI sběrnice, lokální sběrnice) s odlišnými hodinovými signály. Tyto hodinové signály běží navzájem asynchronním způsobem, na straně lokální sběrnice může hodinový signál dosahovat kmitočtu až 50 Hz. Pro podporu dávkových přenosů v tomto asynchronním prostředí je zařízení vybaveno sadou front (FIFOs) o velikosti 128B pro zápisové operace a 64B pro čtecí operace.

Pokročilé funkce PLX jsou ovládány pomocí sady interních registrů přístupných z obou rozhraní. Pro uložení počáteční konfigurace těchto registrů je přítomna paměť EEPROM velikosti až 4kb.

**Režim „J“** zařízení PLX je primárně určen jako komunikační rozhraní pro procesory Intel řady i960. V rámci režimu „J“ může PLX provádět čtyři druhy datových přenosů: přístup ke konfiguračním registrům PLX, zápisové a čtecí operace v režimu PCI Initiator, PCI Target a přenosy DMA. Adresa a data jsou v šířce 32 bitů přenášena sdíleně (anglicky multiplexed address and data). Popis signálů režimu „J“ na straně rozhraní lokální sběrnice je uveden v tabulce 2.3.



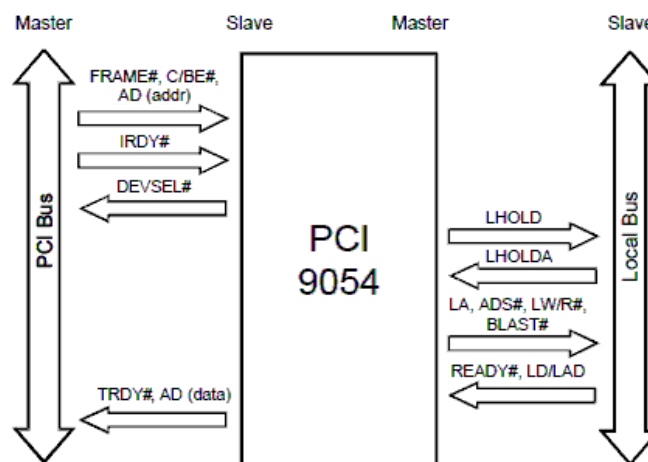
Signál	Název	Směr	Popis
LCLK	Local Clock	I	Hodinový signál z lokální sběrnice
ADS	Address Strobe	I/O	Indikuje platnou adresu na sig. AD
DEN	Data Enable	O	Indikuje platná data na sig. AD
DT/R	Data Transmit/Receive	O	Indikuje, zda jsou data předávána nebo očekávána
LAD	Address Data Bus	I/O	Sdílený signál pro přenos adresy a dat 32 bitů
LHOLD	Local Hold Request	O	Indikuje požadavek PLX na získání kontroly lokální sběrnice
LHOLDA	Local Hold Acknowledge	I	Indikuje předání kontroly nad lokální sběrnici čipu PLX
LW/R	Write/Read	I/O	Rozlišuje čtecí a zápisovou operaci
Ready	Ready	I/O	Strana přenosu Slave indikuje platnost dat
BLAST	Burst Last	I/O	Signál označuje konec dávkového přenosu
LBE	Byte Enables	I/O	Určuje počet platným bytů v rámci datového přenosu
ALE	Address Latch Enable	I/O	Signál je nastaven v průběhu adresové fáze a od-nastaven ještě před začátkem datové fáze

Tabulka 2.3 Signály na rozhraní lokální sběrnice mód „J“

Pro rozlišení typu přenosové transakce je na sběrnici PCI předáváno kódové označení (viz. tabulka 2.1) signálem C/BE[4:0]. Tento signál se však na rozhraní s lokální sběrnici nevyskytuje. Pro rozlišení jednotlivých druhů transakcí při přenosech s lokální sběrnici v režimu PCI „bus master“ proto slouží jeden z vnitřních registrů čipu PLX, jehož obsahem je právě zmíněný kód dle specifikace PCI. Registr je nastavitelný jak ze strany lokální sběrnice, tak ze strany PCI.

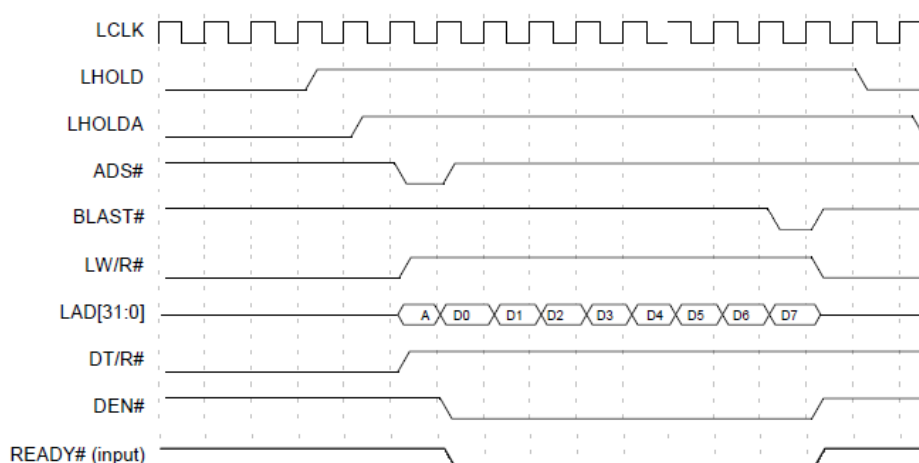
Jak již bylo zmíněno předchozí části kapitoly, vestavěné rozhraní PLX umožňuje čtyři druhy datových přenosů. Jejich přesný průběh je definován v komunikačním protokolu zařízení PLX. Pro účely návrhu vyvíjeného zařízení jsou však významné pouze dva z nich – přenos PCI Initiator a PCI Target. Oba přenosy v sobě zahrnují operaci čtení i zápis, a to jak s jedním datovým cyklem, tak i dávkově (BURST). Zařízení pracující na straně lokální sběrnice bude pro účely této kapitoly nazýváno řadičem lokální sběrnice (zařízení navrhované v rámci této práce je specifickou formou takového řadiče).

PCI Target je druh datového přenosu, ve kterém je činnost vůči lokální sběrnici řízena ze strany PLX. Operace čtení i zápis mají v tomto režimu společnou první fázi komunikace a liší se teprve až od fáze přenosu dat. Obecný princip přenosu PCI Target operace čtení je na obrázku 2.9. Přenosová operace je vždy zahájena nastavením signálu LHOLD ze strany PLX. Řadič lokální sběrnice indikuje její připravenost nastavením signálu LHOLDA. Přijetím signálu LHOLDA zahájí PLX adresovou fázi přenosu. Na datovou linku (AD) je vystavena adresa cílového zařízení a její platnost je potvrzena signálem ADS. Již v tomto hodinovém taktu je signálem LW/R indikováno zda se jedná o čtecí (obr. 2.10) či zápisovou operaci (obr. 2.11). Další průběh přenosu dat se pro obě operace odlišuje.



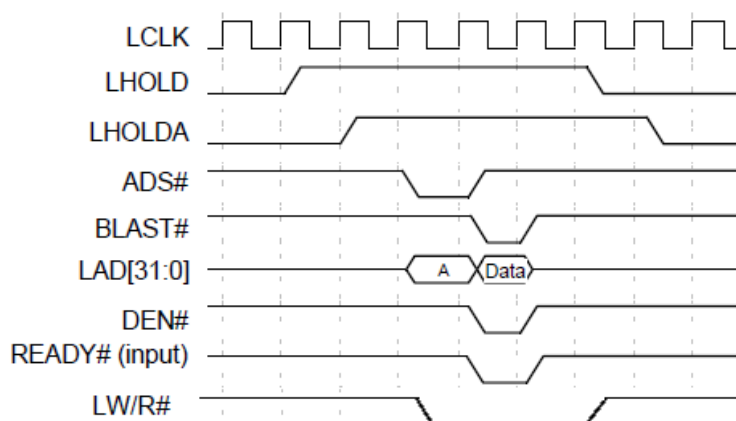
Obrázek 2.9. Princip komunikace při PCI Target čtení  
(převzato z [12])

Navazující datová fáze operace **zápis** dále pokračuje vystavením prvního 32 bitového bloku dat (počet platných dat závisí na signálu LBE). Jejich platnost je potvrzena signálem DEN. Lokální sběrnice dává najevo připravenost k přijetí dat signálem READY. Následně je přijímáno 32 bitů dat v každém taktu hodin lokální sběrnice. Současně se předpokládá, že inkrementace adresy navazujících zapisovaných datových bloků je prováděna v cílovém zařízení lokální sběrnice. Přenos je ukončen nastavením signálu BLAST ze strany PLX při poslední bloku přenášených dat. V případě přenosu pouze jednoho bloku dat (tedy se nejedná o dávkový přenos) je signál BLAST nastaven již při prvním taktu datové fáze. Obě strany přenosové transakce mohou přenos pozastavit vložení tzv. čekacích taktů (wait states). PLX indikuje pozastavení shozením signálu WAIT, řadič sběrnice pak shozením signálu READY.



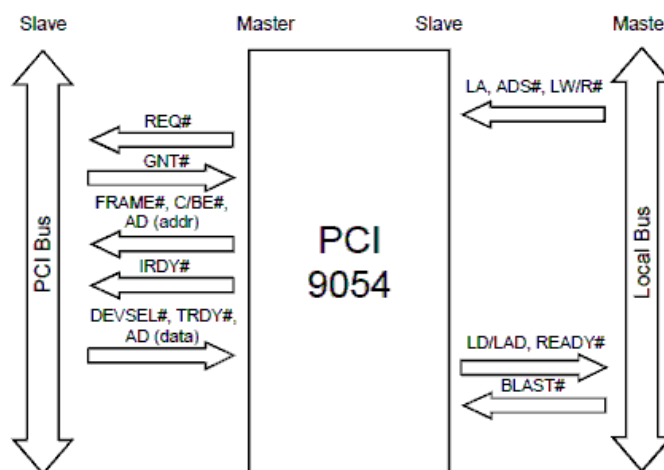
Obrázek 2.10 Operace PCI Target zápis - dávka

Po ukončení adresové fáze a zahájení datové fáze operace **čtení** PLX očekává na datové lince přijetí požadovaných dat. Tuto připravenost indikuje nastavením signálu DEN. PLX je schopno data přijmout již v taktu bezprostředně následujícím adresovou fází přenosu. Jakmile má řadič lokální sběrnice data k dispozici, tyto vystaví na datovou linku a jejich platnost indikuje nastavením signálu READY.



Obrázek 2.11 Operace PCI Target čtení - jednoduchá

V každém dalším taktu jsou pak na datovou linku vystavována navazující data. Inkrementaci cílové adresy je opět úkolem zařízení na lokální sběrnici. Jakmile PLX přijme požadované množství dat, nastaví signál BLAST čímž přenos ukončí. Obě zařízení mohou přenos pozastavit shobením signálů WAIT (PLX) a READY (řadič lokální sběrnice).



Obrázek 2.12 Princip komunikace při PCI Initiator čtení  
(převzato z [12])

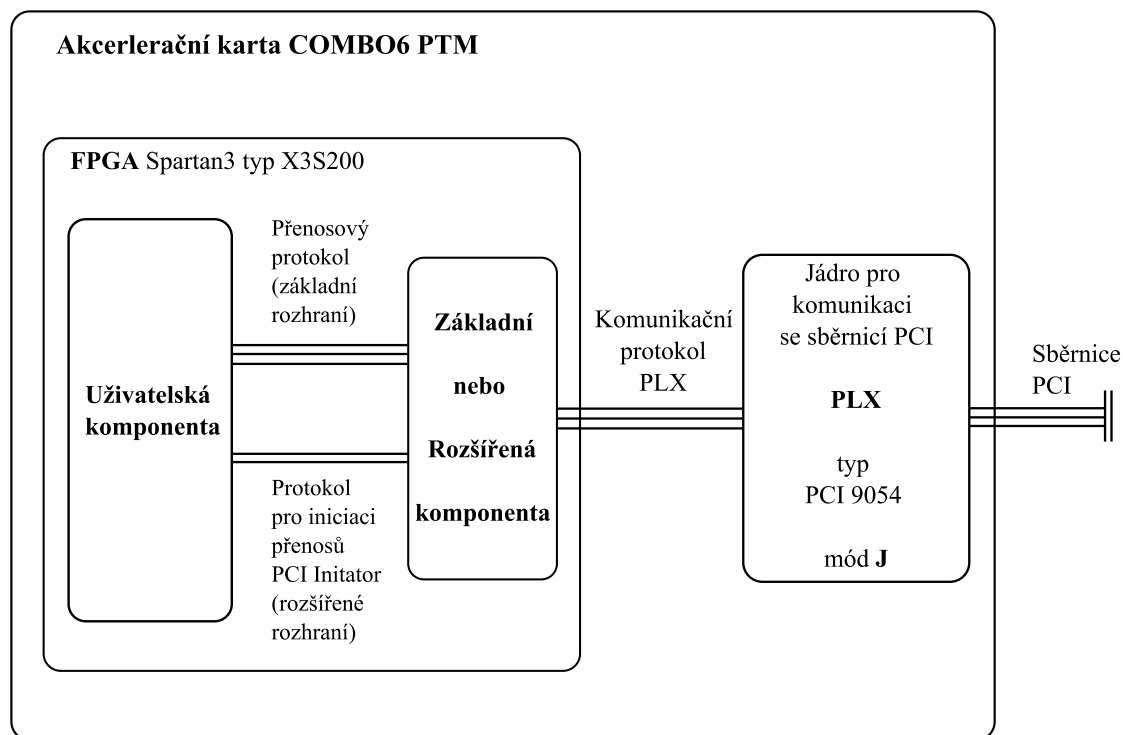
Při datovém přenosu **PCI Initiator** řadič lokální sběrnice přebírá funkci iniciátora prováděných operací a tak nepřímo skrze zařízení PLX funguje jako PCI „bus master“. Příklad operace PCI Initiator je na obrázku 2.12.

Čtecí i zápisová operace u datových přenosů **PCI Initiator** odpovídá svým časovým průběhem včetně nastavení jednotlivých signálů stejným operacím v režimu PCI Target, pouze je prohozena role řadiče lokální sběrnice a PLX a transakce je zahajována odlišným způsobem. Tato totiž nezačíná nastavováním signálů LHOLD a LHOLDA jako v případě PCI Target, nýbrž je zahájena přímo vystavením adresy cílového zařízení na datovou linku a jejím potvrzením signálem ADS. Současně je prohozeny strany u nastavení signálů WAIT a READY určených mimo jiné ke vkládání čekacích taktů.

## 3 Návrh

Kapitola se zabývá návrhem architektury komponenty pro komunikaci s PCI rozhraním reprezentovaným komunikačním protokolem vestavěného jádra PLX. Její umístění je znázorněno na obrázku 3.1. Před návrhem komponenty bylo nutné specifikovat jednoduchý komunikační protokol, který bude použit uvnitř cílového FPGA, a to jak pro komunikaci s navrženou základní ale i rozšířenou komponentou. Jeho struktura včetně časových diagramů je popsána v následující části kapitoly 3.1. Po ní následuje popis architektury základní části komponenty využívající navržený protokol. Základní část komponenty zprostředkovává operace čtení i zápis (jednoduché i dávkové) pro datové přenosy PCI Target.

Pro zachování pokročilých funkcí sběrnice PCI byla vytvořena rozšířená verze komponenty. Tato pracuje v módech PCI Target i PCI Initiator a dalším komponentám uvnitř cílového FPGA zpřístupňuje přenosy DMA (jednoduché i dávkové). Popisu rozšířené komponenty se věnuje závěrečná část kapitoly.



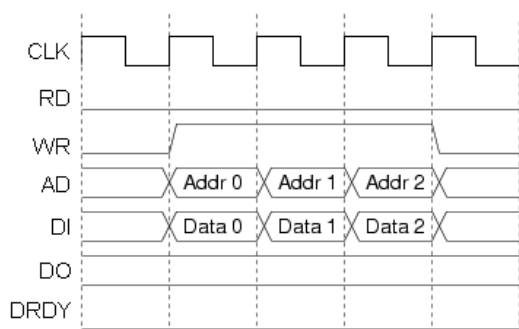
Obrázek 3.1 Schéma umístění navrhovaných komponent

### 3.1 Komunikační protokol uvnitř FPGA

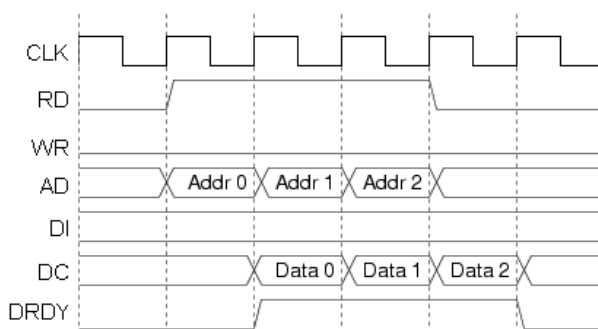
Komunikační protokol uvnitř FPGA se skládá z protokolu pro komunikaci skrze základní rozhraní a z protokolu pro iniciaci přenosů v režimu PCI Initiator (DMA přenosy). Oba protokoly byly navrženy s ohledem na komunikační protokol akceleračního jádra PLX.

Protokol pro komunikaci přes základní komunikační rozhraní (**přenosový protokol**) je využíván u **základní i rozšířené verze navržené komponenty** a umožňuje provedení dvou operací – zápis do a čtení z uživatelské komponenty. Protokol využívá signálů RD (indikuje operaci čtení), WR (indikuje operaci zápis), AD (adresa místa ke čtení či zápisu, 32b), DI (data z navržené do uživatelské komponenty, 32b), DO (data z uživatelské do navržené komponenty) a signálu DRDY (uživatelská komponenta indikuje platnost vystavených přečtených dat).

**Zápisová operace** (obr 3.2) - v rámci jednoho taktu hodinového signálu je vystavena platná adresa (na portu AD, 32 bitů), současně s adresou jsou vystavena data pro zápis (na portu DI, 32 bitů) a celá zápisová operace je řízena signálem WR. Uživatelská komponenta by měla reagovat na tento protokol zápisem dat (na portu DI) do registru nebo paměti na adrese AD.

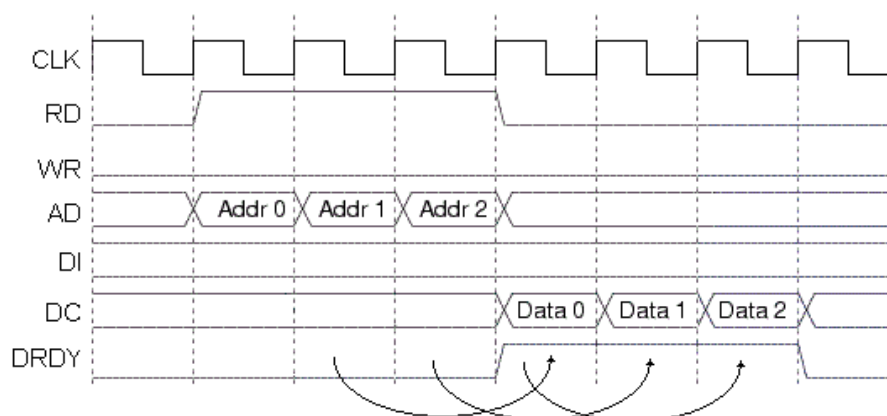


Obr. 3.2 Zápisová operace



Obr. 3.3 Čtecí operace

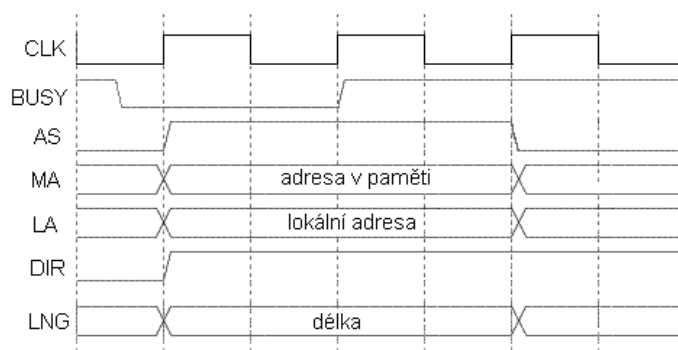
**Čtecí operace** (obr 3.3) - podobně jako v předchozím případě je na sběrnici vystavena platná adresa (na portu AD) a současně s adresou je nastaven řídicí signál RD. Uživatelská komponenta by měla reagovat na tuto operaci tak, že přečte požadovaná data na adrese AD a vystaví je na výstupní port DO (32 bitů). Výstupní data jsou požadována až v následujícím hodinovém taktu a s platnými daty je nastaven také signál DRDY (Data Ready). Uživatelská komponenta tento signál aktivuje společně s platnými daty na portu DO nejdříve jeden takt za signálem RD. Pokud není uživatelská komponenta schopna data vystavit do následujícího hodinového taktu, je možné je vystavit později, ale musí být současně také zpožděn signál DRDY. Signál DRDY nesmí být zpožděn o více než deset hodinových taktů. Data i u zpožděného přenosu musí být vystavována kontinuálně, není tedy možné vkládat tzv. čekací takty uprostřed již započatého datového přenosu (obr 3.4).



Obr. 3.4 Čtecí operace se zpožděním realizovaným signálem DRDY (zpoždění 2 takty)

V případě, že jsou signály *RD* nebo *WR* jsou vystaveny po dobu více hodinových taktů, jedná se dávkovou operaci. Adresa na portu *ADDR* je postupně inkrementována a na datových portech *DI* a *DO* jsou vystavovány jednotlivé 32b datové bloky.

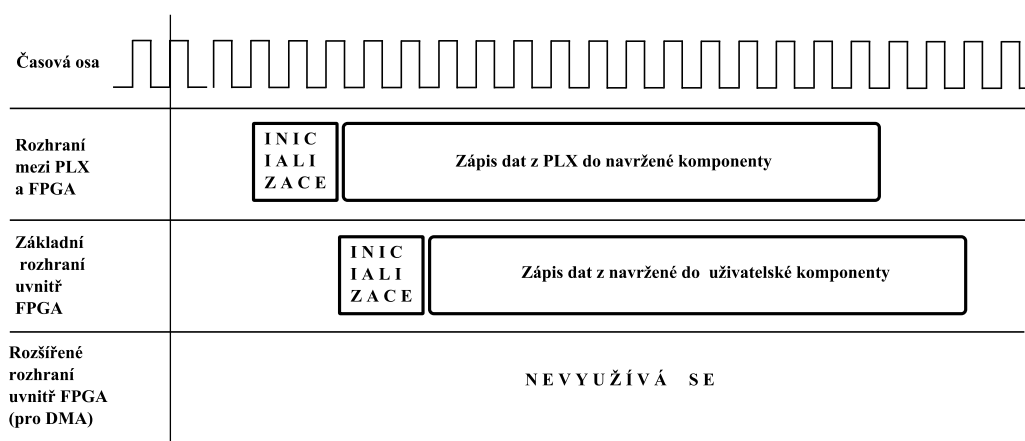
**Protokol pro iniciaci přenosů v režimu PCI Initiator** využívá rozšiřujícího rozhraní, které je přítomno pouze u **rozšířené komponenty**. Rozhraní je tvořeno signály LA (adresa paměti v zařízení v FPGA, 32 bitů), MA (adresa v paměti mimo FPGA, 32 bitů), LNG (šířka 10 bitů), DIR (směr operace), BUSY a AS (address strobe). Signálem LNG určuje uživatelská komponenta počet 32b bloků přenášených dat, tedy délku přenosu. V případě, že je LNG nastaveno na hodnotu jedna, jedná se o operaci přenesení jednoho bloku dat („SINGLE“), jinak se jedná o operaci dávkovou („BURST“). Vzhledem k šířce signálu je maximální délka dávkového přenosu  $2^{10}$ , tedy během jedné dávkové operace lze přenést maximálně 1024 32b bloků dat (tj. 4 KB). Připravenost zařízení k zahájení operace v režimu PCI Initiator je indikována signálem BUSY (připraven v případě hodnoty 0). Platnost vystavených adres ze strany uživatelské komponenty potvrzuje signál AS, který indikuje také pokyn k zahájení operace. Zahájení operace je pak potvrzeno nastavením signálu BUSY. Nepřipravenost k operaci je způsobena probíhající operací v režimu PCI Target či PCI Initiator.



Obrázek 3.4 Inicie přenosu v režimu PCI Initiator – pouze rozšiřující část rozhraní

## 3.2 Časová souslednost užití protokolů v rámci operací

**Operace PCI Target Write** (obr. 3.5) – v průběhu jsou využívány pouze dva ze tří přítomných protokolů. Inicializace přenosu nastává ze strany PLX, které nastavením řídicího signálu (LHOLD) vyzývá navrženou komponentu k zahájení přenosu. V průběhu inicializace dojde k potvrzení zahájení přenosu (LHOLDA) a v dalším hodinovém taktu je navržené komponentě předána cílová adresa prvního 32b bloku dat. Jakmile je adresa převzata, dochází k inicializaci přenosu zápis také na straně uživatelské komponenty. K inicializaci dochází pouze na základním rozhraní. Po ukončení inicializace jsou pak kontinuálně předávána data z PLX skrze navrženou až do uživatelské komponenty. Ze strany PLX je v úvodu předána pouze počáteční adresa, na straně s uživatelskou komponentou je však ke každému bloku dat vystavena vždy platná adresa, která je z tohoto důvodu postupně vždy inkrementována. Přenos je ukončen jádrem PLX pomocí signálu BLAST.

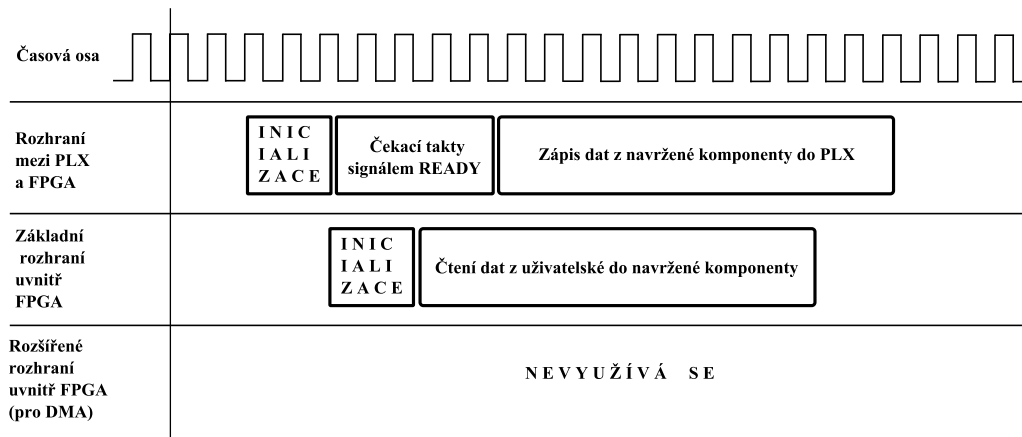


Obrázek 3.5 Průběh operace PCI Target Write

**Operace PCI Target Read** (obr. 3.6) – i zde jsou využívány pouze dva protokoly. Přenos je inicializován obdobně jako u operace PCI Target Write. Po ukončení inicializace, tedy po předání adresy prvního bloku čtených dat, je zahájena inicializace přenosu čtení na straně základního rozhraní s uživatelskou komponentou. V době mezi inicializací a vystavením načtených dat je činnost na rozhraní z PLX pozastavena vložení čekacích stavů (realizováno signálem READY). Počet čekacích stavů závisí především na době, za kterou je schopna uživatelská komponenta požadovaná data načíst a vystavit na příslušném rozhraní. Počet čekacích stavů nesmí překročit 16 hodinových taktů. Přenos je ukončen ze strany PLX vystavením signálu BLAST u posledního 32b bloku čtených dat. Stejně jako u předchozí operace je adresa na základním rozhraní s uživatelskou komponentou vždy postupně inkrementována.

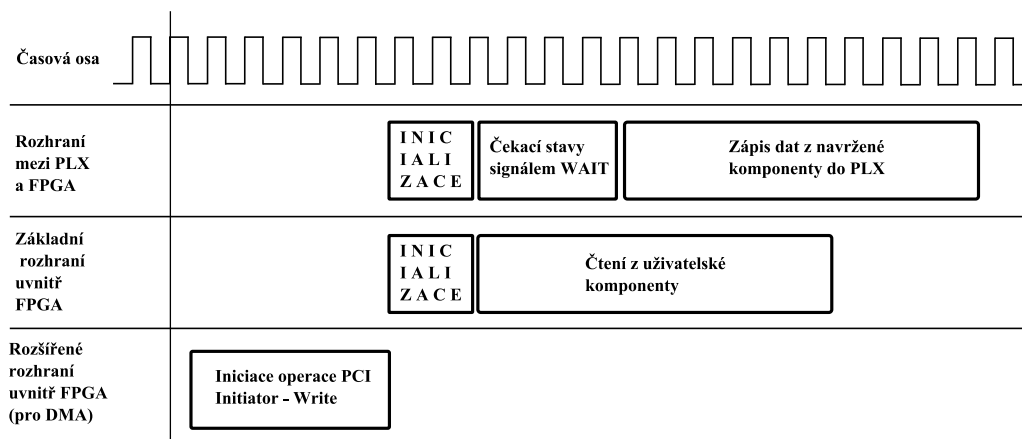
Počet zapisovaných a čtených 32b bloků dat není nijak omezen - u operací PCI Target stanoven vždy ze strany akceleračního jádra PLX.





Obrázek 3.6 Průběh operace PCI Target Read

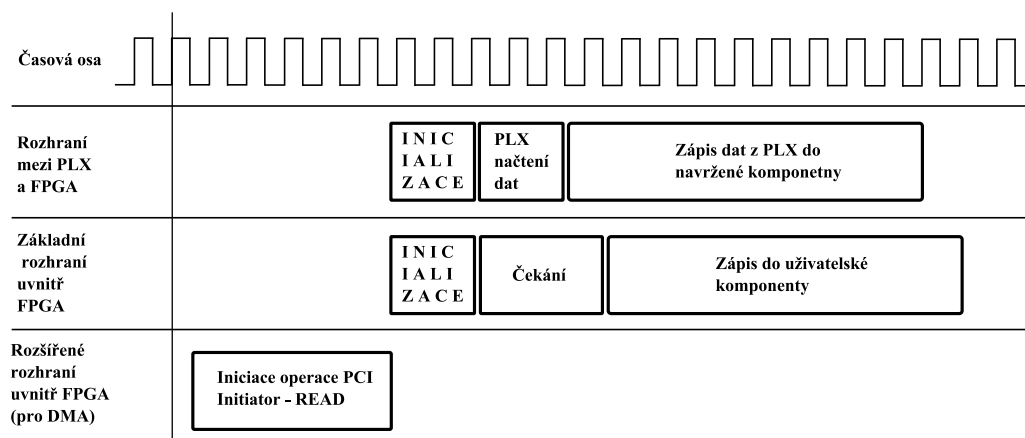
**Operace PCI Initiator Write** (obr. 3.7) – tato je zahájena skrze rozšířené rozhraní uživatelskou komponentou. V průběhu inicializace operace na tomto rozhraní dojde k uložení všech atributů transakce (adresa v uživatelské komponentě, adresa v globální paměti, délka přenosu, směr přenosu). Jakmile je tento proces dokončen je současně zahájena operace čtení z uživatelské komponenty a iniciován přenos skrze jádro PLX. Jádro PLX dává připravenost k operaci signálem READY. Nejsou-li v této době ještě k dispozici data načtená z uživatelské komponenty, jsou do přenosu s PLX vloženy čekací stavy signálem WAIT. Akcelerační jádro PLX může v průběhu přenosu také vkládat čekací stavy a to shozením signálu READY. V tomto případě je čtení dat z uživatelské komponenty na příslušnou dobu pozastaveno. Přenos je ukončen vystavením posledního přenášeného 32b bloku dat a nastavením signálu BLAST.



Obrázek 3.7 Průběh operace PCI Initiator Write

Při inicializaci přenosu skrze rozšířené rozhraní jsou předány pouze počáteční adresy, na straně základního rozhraní s uživatelskou komponentou je tedy adresa postupně vždy inkrementována. Na straně s PLX je předána cílová adresa pouze jednou, a to ještě v průběhu inicializace tohoto rozhraní.

**Operace PCI Initiator Read** (obr. 3.8) – stejně jako předchozí operace je zahájena inicializací ze strany uživatelské komponenty skrze rozšířené rozhraní za současného uložení všech atributů přenosu. Následuje předání požadavku na čtení dat akceleračnímu jádru PLX a příprava k uložení takto načtených dat skrze základní rozhraní uživatelské komponenty. Dokud však nejsou připravena načtená data u rozhraní s PLX nedojde k zahájení tohoto zápisu - není nastaven signál WR (na obrázku je označeno jako „čekání“). Jádro PLX indikuje připravenost k předání dat signálem READY. Jakmile je tento nastaven, dojde ke kontinuálnímu předávání 32b datových bloků z PLX skrze navrženou komponentu do uživatelské komponenty.



Obrázek 3.8 Průběh operace PCI Initiator Read

V průběhu přenosu dat mohou být na rozhraní s PLX vkládány čekací stavy a to jak ze strany PLX (shoením signálu READY) tak ze strany uživatelské komponenty (shoením signálu WAIT). Ukončení operace je jádru PLX indikováno signálem BLAST. Počet čekacích stavů způsobených signálem WAIT nesmí překročit 16.

Při inicializaci přenosu skrze rozšířené rozhraní jsou opět předány pouze počáteční adresy, na straně základního rozhraní s uživatelskou komponentou je tedy adresa postupně vždy inkrementována. Na straně s PLX je předána globální adresa pouze jednou a to ještě v průběhu inicializace tohoto rozhraní.

V případě, že dojde k **současnému požadavku** na přenos v režimu PCI Target i PCI Initiator (tedy současný požadavek ze strany PLX i uživatelské komponenty) je upřednostněn přenos PCI Target.

### 3.3 Architektura základní komponenty

Návrh základní komponenty byl veden s cílem minimalizovat nároky na potřebné hardwarové prostředky programovatelného pole FPGA, které tak mohou být využity uživatelskou komponentou. Celý systém je řízen skrze rozhraní PLX, komponenta tedy pracuje pouze v režimu PCI Target. V úvodu kapitoly je uveden soupis všech užitých signálů (tab. 3.1). Funkce signálů na obou rozhraních byla popsána v předcházejících kapitolách, funkce interních signálů bude vysvětlena v dalších částech této kapitoly. Po soupisu signálů následuje popis blokového schématu komponenty. Závěr je věnován vysvětlení funkce centrálního řídicího systému – konečného automatu.

Název	Směr
<b>Obecné</b>	
CLK	I
RESET	I
<b>Rozhraní s PLX</b>	
LHOLD	I
LHOLDA	O
ADS	I
BLAST	I
LAD(32)	I/O
DEN	I
LWR	I
READY	O

Tabulka 3.1a Signály základní komponenty

Název	Směr
<b>Základní rozhraní</b>	
RD	O
WR	O
DRDY	I
AD(32)	O
DI(32)	O
DO(32)	I
<b>Interní signály</b>	
ad_we	X
mux_to_ad	X
ad_to_mux	X
trist_sig	X

Tabulka 3.1b Pokračování

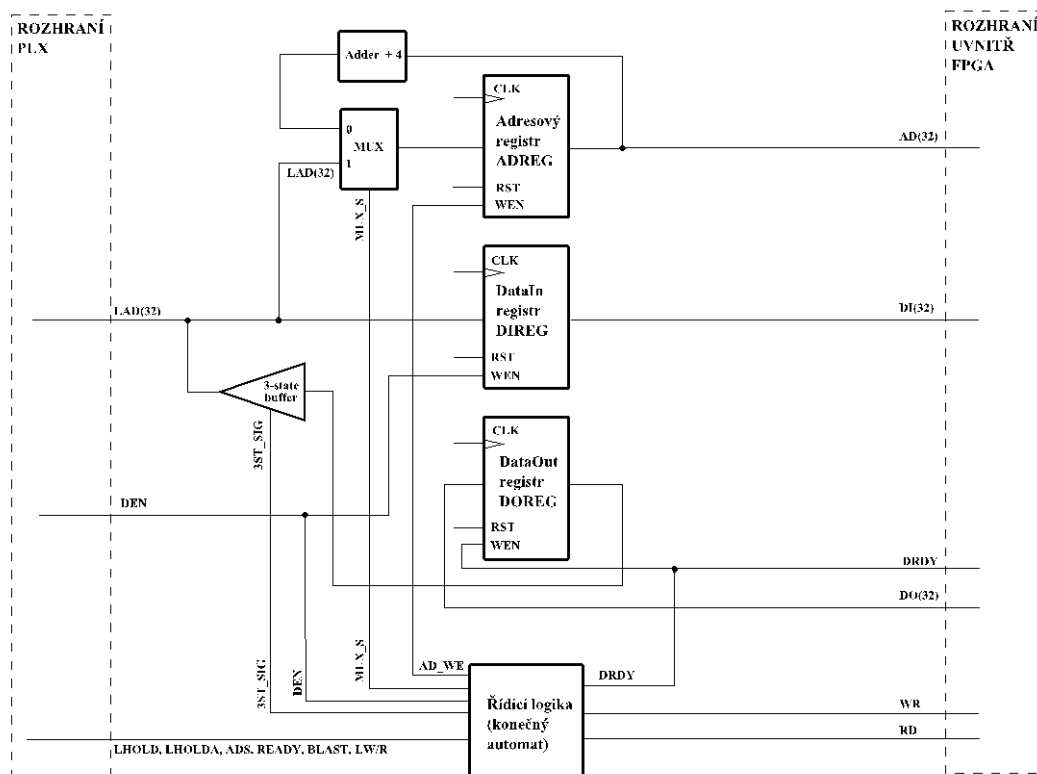
V základní komponentě nebyly z důvodu nepotřebnosti využity signály DT/R (funkčně doplňuje signál DEN), LBE (datová šířka je vždy 32b) a ALE (funkčně doplňuje signál ADS). Současně nebyl využit signál WAIT, neboť základní komponenta neumožňuje vkládání čekacích taktů ze strany PLX.

Z funkčního pohledu je **základní komponenta** složena ze dvou logických celků – **registrového pole** a **centrálního řídicího systému** – konečného automatu. Vzájemné propojení obou celků včetně příslušných signálů základní komponenty je znázorněno na obrázku 3.9.

**Registrové pole** slouží jako obousměrné centrální adresové a datové překladiště a je tvořeno

- **Datovým registrem DIREG** – registr šířky 32b s asynchronním resetem a povolením zápisu. Slouží k uložení dat vystavených z PLX sběrnici LAD a zapisovaných do uživatelské komponenty, povolení zápisu do registru je řízeno signálem DEN,

- **datovým registrem DOREG** – registr šířky 32b s asynchronním resetem a povolením zápisu. Slouží k uložení dat načtených z uživatelské komponenty a jejich následnému předání jádru PLX skrze sběrnici LAD. Povolení k zápisu je řízeno signálem DRDY,
- **adresovým registrem ADREG** – registr šířky 32b s asynchronním resetem a povolením zápisu. Povolení k zápisu obsluhuje centrální řídicí systém (ad\_we). Adresový registr je doplněn 32b sčítačkou a multiplexorem 2x1 o šířce 32b. Úkolem sčítačky je inkrementovat předávanou adresu v rámci dávkových přenosů. Multiplexor na základě signálu řídicí logiky mux\_x rozlišuje, zda je adresový registr plněn výstupem sčítačky (ad\_to\_mux – inkrementace adresy) či sběrnici LAD (nahrávání adresy).
- **3-stavový zesilovač signálu („buffer“)** – obsluhuje obousměrnou sběrnici LAD, řízen řídicí logikou pomocí signálu trist\_sig. V případě operace zápis se nachází ve stavu vysoké impedance. Pouze v části operace čtení tento stav mění a na sběrnici LAD předává obsah registru DOREG.



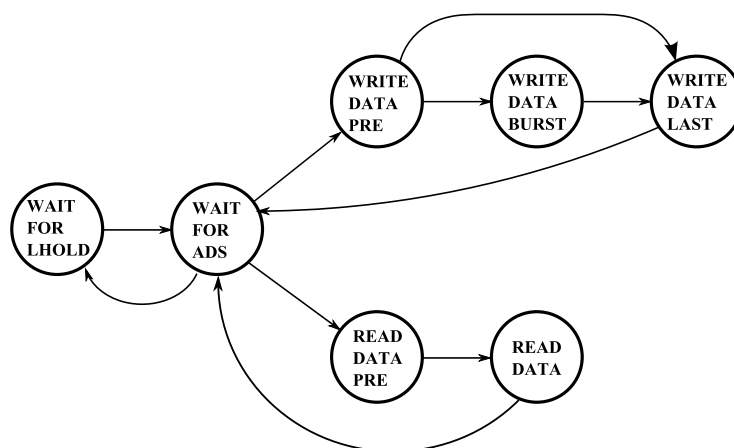
Obrázek 3.9: Blokové schéma základní komponenty

**Centrální řídicí systém (Řídicí logika)** ovládá ostatní části základní komponenty, v souladu s komunikačními protokoly na obou rozhraních nastavuje interní a výstupní signály. Je navržen na principu synchronního konečného automatu Mealyho typu s asynchronním resetem, tj. nastavení provádí v závislosti na vstupních signálech a současném stavu.

Konečný automat řídicí logiky má následujících 8 stavů:

- WAIT FOR LHOLD,
- WAIT FOR ADS,
- WRITE DATA PRE,
- WRITE DATA BURST,
- WRITE DATA LAST,
- READ DATA PRE,
- READ DATA.

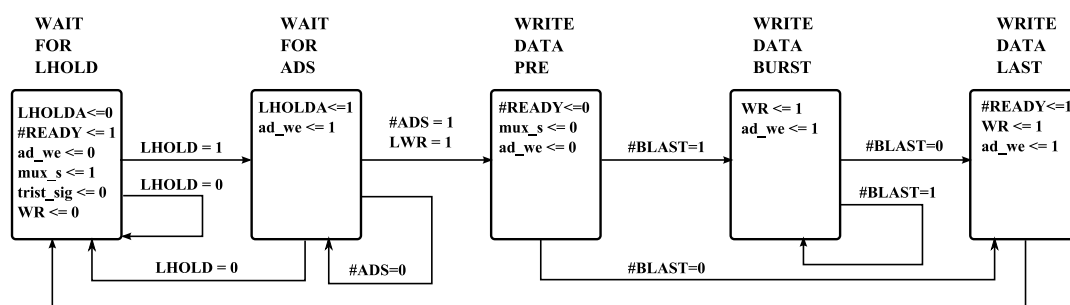
Obecný graf návaznosti stavů je zobrazen na obrázku 3.10. Jak je patrné, dva stavy jsou sdíleny, operace zápis sestává z 3 individuálních stavů a operace čtení pak ze dvou. Počátečním stavem konečného automatu je stav WAIT\_FOR\_LHOLD, po shození signálu RESET se automat nachází právě v tomto stavu.



Obrázek 3.10 Graf přechodů konečného automatu

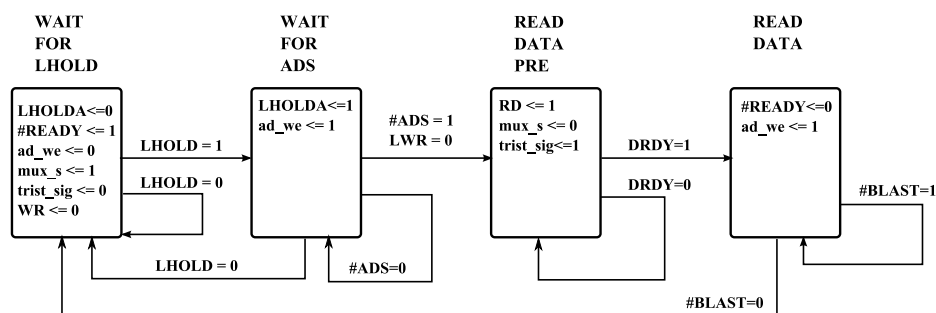
**Operace Zápis** (PCI Target Write) je realizována posloupností čtyř stavů v případě přenosu jednoho datového bloku a pěti stavů v případě dávkového přenosu. Konečný automat setrvává v počátečním stavu (WAIT\_FOR\_LHOLD) až do případného nastavení požadavku na přenos jádrem PLX na vodiči LHOLD. Konečný automat indikuje svou připravenost signálem LHOLDA a přechází do stavu očekávání adresy (WAIT\_FOR\_ADS). Současně nastavuje plnění adresového registru (mux\_s) a povoluje jeho zápis (ad\_we). Jakmile je vystavena platná adresa (ADS) a požadovaná operace je zápis (LWR) konečný automat potvrdí zahájení operace (READY) a adresový registr uvede do režimu zvyšování adresy (WAIT\_DATA\_PRE). Obsah

registrů DIREG a ADREG je následně předáván uživatelské komponentě (WAIT\_DATA\_BURST), operace je indikována signálem WR. Data i adresa jsou uživatelské komponentě předávána se zpožděním jednoho taktu. Po obdržení posledního bloku dat (BLAST) automat operaci ukončí (WRITE\_DATA\_LAST) a vrací se do stavu čekání na adresu (WAIT\_FOR\_ADS). Jedná-li se o jednoduchý přenos, tzn. první datový blok je obdržen společně se signálem BLAST, automat přechází přímo do stavu WRITE\_DATA\_LAST. Úplný diagram průběhu operace zápis je na obázku 3.11.



Obrázek 3.11 Diagram stavů konečného automatu u operace zápis (PCI Target Write)

**Operace Čtení** (PCI Target Read) sestává z průchodu čtyř stavů řídicího konečného automatu. Inicializace operace probíhá obdobně jako u operace zápis. Činnost automatu se liší teprve při převzetí platné adresy (ADS) spolu s požadavkem na operaci čtení (LWR). Na základě této kombinace je uživatelské komponentě vystaven požadavek na čtení (RD) společně s počáteční adresou (READ\_DATA\_PRE). V následujících taktech je předáváná adresa postupně



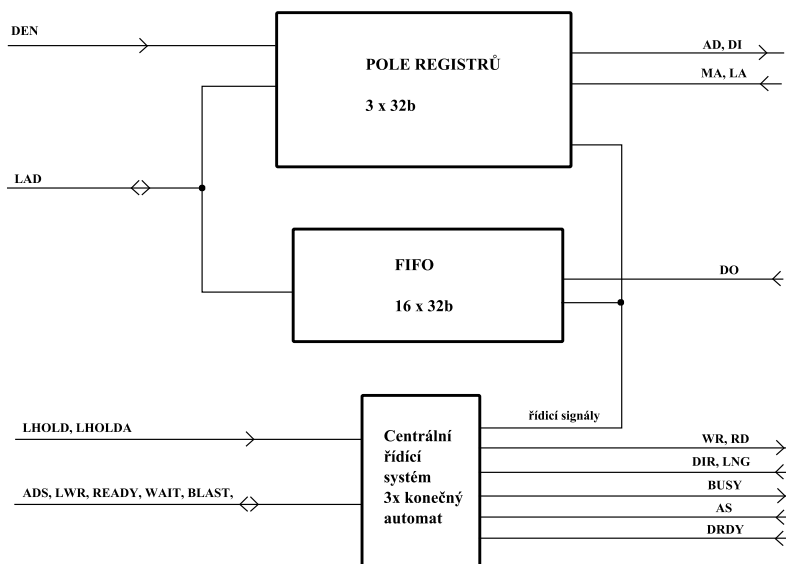
Obrázek 3.12 Diagram stavů konečného automatu u operace čtení (PCI Target Read)

inkrementována, ale na rozhraní s PLX jsou zatím vkládány pouze čekací takty. Jakmile uživatelská komponenta načte a vystaví první platná data (DRDY), tato jsou předána sběrnici LAD jádru PLX. Jejich platnost je potvrzena signálem READY (READ\_DATA). V tomto stavu

konečný automat zůstává až do ukončení transakce, tj. předání posledního 32b bloku dat (indikováno BLAST ze strany PLX). Nadbytečně přečtená data z uživatelské komponenty jsou zahozena a automat přechází do stavu čekání na adresu (WAIT\_FOR\_ADS). Počet nadbytečně načtených dat odpovídá zpoždění mezi vystavením požadavku na čtení a prvními platnými daty vrácenými uživatelskou komponentou + 1. Diagram přechodu stavů konečného automatu v průběhu operace zápis je na obrázku 3.12.

### 3.4 Architektura rozšířené komponenty

Architektura rozšířené komponenty (obr. 3.13) byla navržena s cílem zprostředkovat připojené uživatelské komponentě umístěné ve stejném FPGA přenosy v režimu PCI Initiator (režim „master“) i PCI Target (režim „slave“). Celý systém je tedy řízen buď skrze rozhraní PLX, nebo skrze rozšířené rozhraní uživatelskou komponentou. Současně byla zachována možnost vkládání čekacích stavů na začátku i v průběhu přenosových transakcí ze strany PLX i uživatelské komponenty. V úvodu kapitoly je uveden soupis vstup/výstupních a interních řídicích signálů (tab. 3.2). Funkce signálů na obou rozhraních byla popsána v předcházejících kapitolách, funkce interních řídicích signálů bude vysvětlena v dalších částech této kapitoly. Po soupisu signálů následuje popis blokového schématu komponenty rozdělený na jednotlivé funkční celky. Závěr je věnován vysvětlení funkce centrálního řídicího systému sestávajícího z jednoho hlavního a dvou pomocných konečných automatů a je zakončen popisem funkce jednotlivých automatů v průběhu všech operací. Úplné schéma rozšířené komponenty se nachází v příloze D práce.



Obrázek 3.13 Zjednodušené schéma rozšířené komponenty

Následující tabulka obsahuje veškeré vstup/výstupní signály komponenty na všech rozhraních (PLX, základní, rozšířené) a interní řídicí signály. Nejsou zde pro přehlednost zahrnuty veškeré interní propojovací signály, jejich užití a umístění je patrné z úplného schématu rozšířené komponenty. Ani v rozšířené komponentě nebyly z důvodu nepotřebnosti využity signály DT/R (funkčně doplňuje signál DEN), LBE (datová šířka je vždy 32b) a ALE (funkčně doplňuje signál ADS).

Název	Směr	Název	Směr	Název	Směr
<b>Obecné</b>		<b>Základní rozhraní</b>		<b>Interní říd. signály</b>	
CLK	I	RD	O	initiator_target	X
RESET	I	WR	O	increase_load	X
		DRDY	I	main_sender	X
<b>Rozhraní s PLX</b>		AD	O	main_filler_FSM	X
LHOLD	I	DI	O	lad_fifo	X
LHOLDA	O	DO	I	address_data	X
		<b>Rozšířené rozhraní</b>			
ADS	I/O	BUSY	O	fill	X
BLAST	I/O	AS	I	send	X
LAD(32)	I/O	DIR	I	sent	X
DEN	I	MA(32)	I	fiforen	X
LWR	I/O	LA(32)	I	fifowen	X
READY	I/O	LEN(10)	I	fifoamfull	X
WAIT	I/O			fifoempty	X

Tabulka 3.2 Vybrané signály rozšířené komponenty

Z funkčního pohledu je **rozšířená komponenta** (obr. 3.12) složena ze tří logických celků – **pole tří registrů, fronty (FIFO) a centrálního řídicího systému (řídicí logiky)** – tří konečných automatů (hlavní MAIN, pomocný SENDER a FILLER).

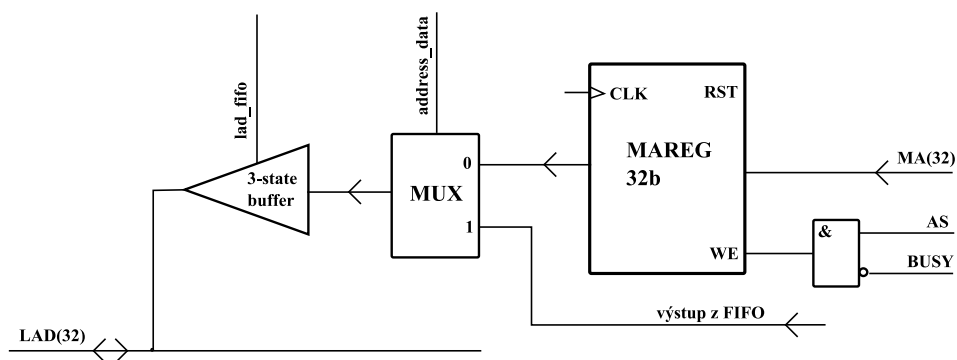
**Registrové pole (3 x 32b)** slouží jako obousměrné centrální adresové a datové uložisko a je tvořeno

- registrem pro uložení dat ve směru PLX->uživatelská komponenta – **DIREG**,
- registrem adresy mimo FPGA – **MAREG**,
- registrem adresy v FPGA – **LAREG**.

Registr **DIREG** je široký 32b s asynchronním resetem a povolením zápisu. Slouží k uložení dat vystavených z PLX sběrnici LAD a zapisovaných do uživatelské komponenty. Povolení zápisu do registru je řízeno signálem DEN. Jeho vstup je napojen přímo na sběrnici LAD, výstup pak přímo na datový vstup uživatelské komponenty DI. Registr je využíván u operací PCI Target Write a PCI Initiator Read.

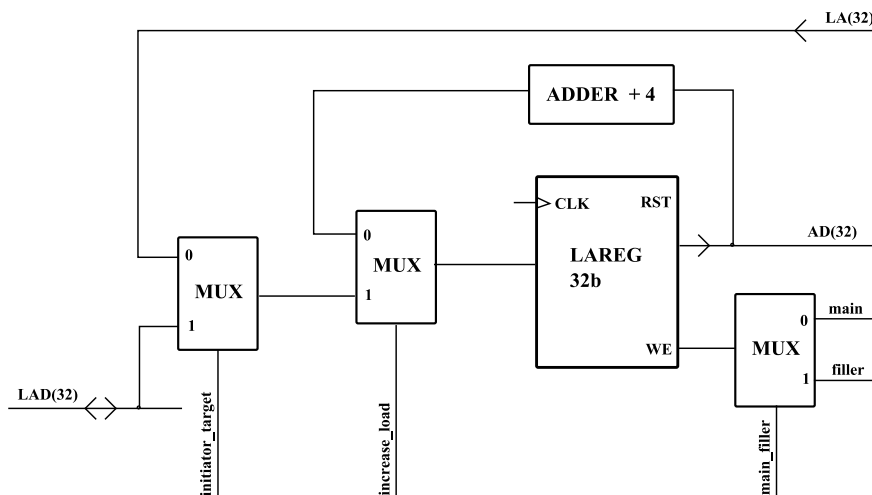


Registr **MAREG** (obr. 3.14) je šířky 32b s asynchronním resetem a povolením zápisu. Slouží k uložení zdrojové/cílové adresy paměťového místa mimo FPGA u operací PCI Initiator (např. operační paměť RAM apod.). Zápis do registru je povolen v době iniciace přenosu PCI Initiator ze strany uživatelské komponenty skrze rozšířené rozhraní (aktivní signál AS, rozšířené zařízení nepracuje – neaktivní BUSY). Předání obsahu registru na sběrnici LAD je řízeno centrálním řídicím systémem (signály address\_data, lad\_fifo) a nastává v době zahájení operace na rozhraní s PLX.



Obrázek 3.14 Schéma zapojení registru MAREG

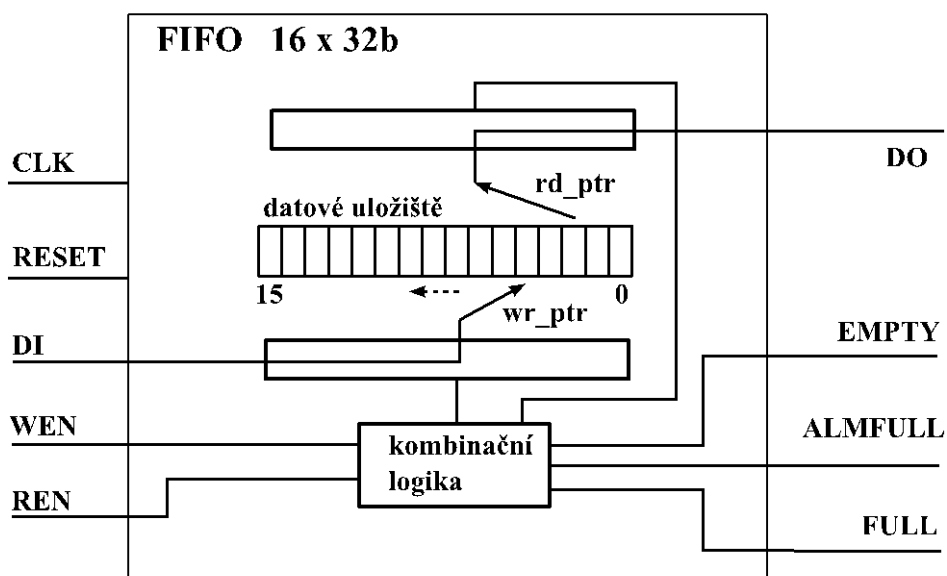
Registr **LAREG** (obr. 3.15) je šířky 32b s asynchronním resetem a povolením zápisu. Slouží k uložení zdrojové/cílové adresy paměťového místa uvnitř FPGA u všech operací. Zápis do registru je kontrolován centrálním řídicím systémem, a to buď jeho hlavní (konečný automat



Obrázek 3.15 Schéma zapojení registru LAREG

MAIN) nebo pomocnou částí (konečný automat FILLER). Registr pracuje v režimu přímého plnění jeho obsahu nebo v režimu postupného zvyšování adresy (increase\_load). Při přímém plnění jsou v operacích PCI Target ukládána data ze zařízení PLX (LAD), u operací PCI Initiator pak data z uživatelské komponenty (LA). Veškeré řídicí signály jsou nastavovány hlavní částí centrálního řídicího systému. U operací PCI Target Write a PCI Initiator Read je také celý systém spravován pouze hlavní částí. V průběhu operací PCI Target Read a PCI Initiator Write je řízení registru předáno pomocnému automatu FILLER (main\_filler), tento však ovládá pouze postupné zvyšování/nezvyšování adresy pomocí střídavého povolování zápisu.

**Fronta FIFO** (z angl. „First In First Out) slouží jako datové překladiště pro data směřující z uživatelské komponenty do PLX. Je určena pro operace PCI Target Read a PCI Initiator Write. Pracuje na synchronním principu s asynchronním resetem a je schopna pojmout až šestnáct 32b bloků dat. Jejím hlavním úkolem je řešení čekacích stavů ze strany PLX. V době vložení čekacího stavu ze strany PLX jsou data z uživatelské komponenty již před načtena a do doby pokračování v transakci je nutno tyto prozatímně uložit. Současně svou funkcí eliminuje krátkodobé výpadky v dodání dat ze strany uživatelské komponenty. Počet před načtených dat je závislý na zpoždění mezi žádostí na čtení a následném vystavení platných dat uživatelskou komponentou a nesmí překročit 10 taktů.



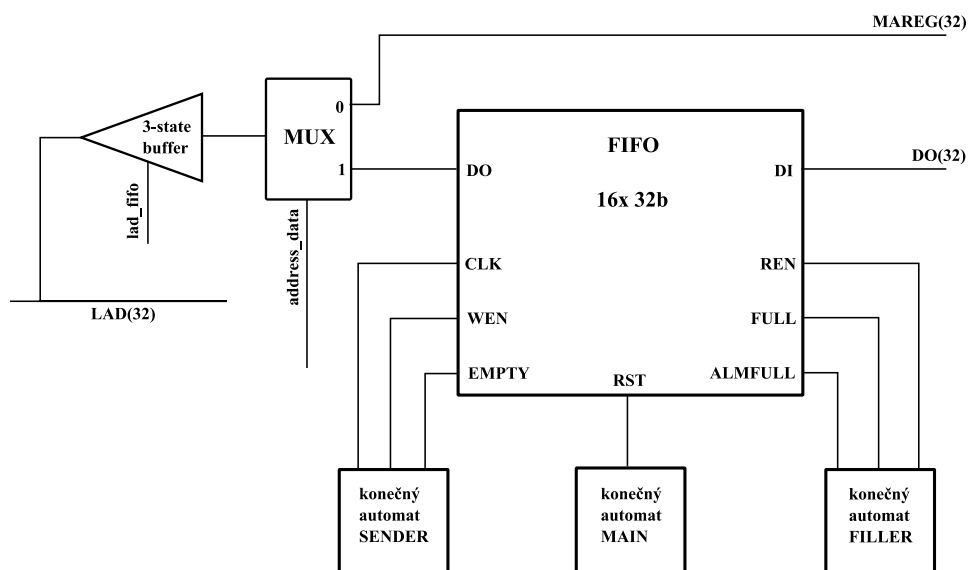
Obrázek 3.16 Fronta FIFO

Fronta je složena z dvouportové distribuované paměti RAM a kombinační logiky zpracovávající vstupní signály. Zápis do fronty je řízen pomocí signálu WEN, čtení pak signálem REN. Svůj stav fronta signalizuje pomocí tří výstupních signálů

- EMPTY – fronta je prázdná
- FULL – fronta je plná
- ALMFULL – fronta obsahuje „prázdné“ položky postačující pouze k uložení již vyžádaných dat z uživatelské komponenty, v této době by mělo dojít k ukončení požadavku na čtení z uživatelské komponenty (RD)

V době kdy není fronta prázdná nebo ve stavu reset, jsou na výstupu (DO) vystavena vždy první platná data.

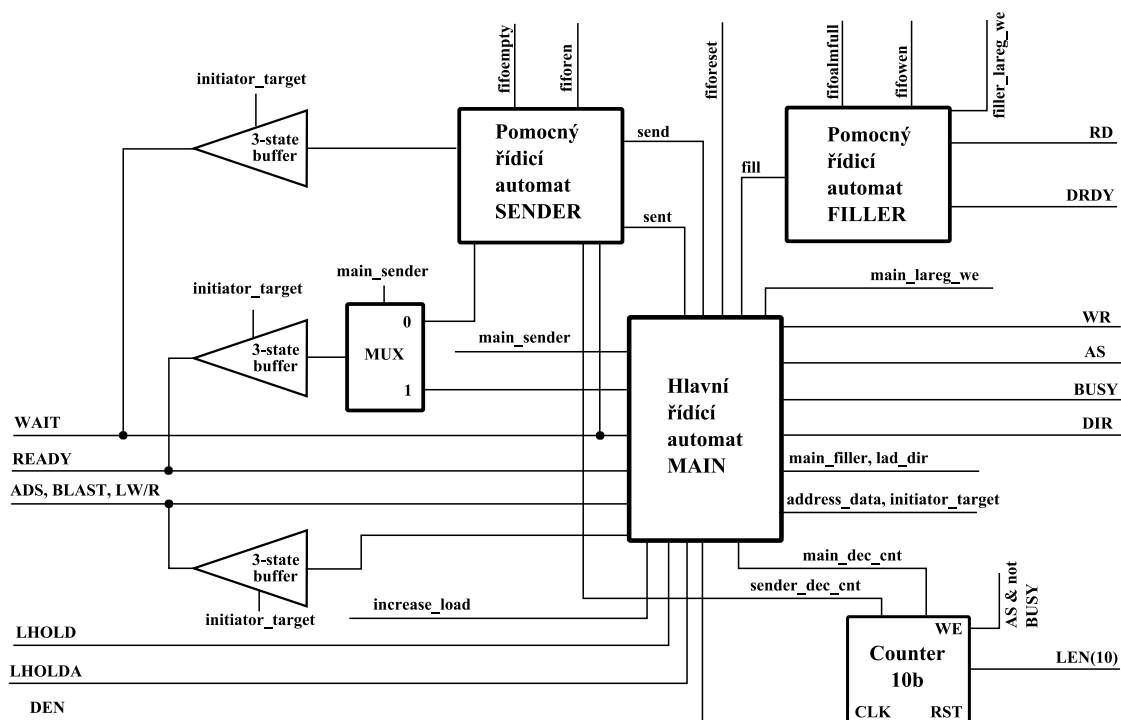
Schéma **zapojení fronty v rámci rozšířené komponenty** je zobrazeno na obrázku 3.17. O ovládání fronty uvnitř rozšířené komponenty se v rámci operací se starají pomocné konečné automaty centrálního řídicího systému - SENDER (řízení čtení z fronty) a FILLER (řízení zápisu do fronty). Výstup z fronty je za předpokladu správného nastavení interních řídicích signálů (address\_data, lad\_fifo) předáván na sběrnici LAD. Po dokončení operace je obsah fronty vždy uveden do počátečního stavu signálem reset řízeným hlavním automatem logiky - MAIN.



Obrázek 3.17 Schéma zapojení komponenty fronta (FIFO)

**Centrální řídicí systém (řídicí logika)** sestává ze tří konečných automatů – hlavního (MAIN) a dvou pomocných (SENDER, FILLER). Hlavní automat MAIN provádí nastavení

většiny interních řídicích signálů, uvádí do provozu a ukončuje činnost obou pomocných automatů. Je činný u všech operací. Pomocný automat FILLER provádí kontinuální plnění fronty na základě její obsazenosti, přičemž sám řídí čtení z uživatelské komponenty včetně obsluhy inkrementace registru LAREG. Je činný pouze u operací PCI Target Read a PCI Initiator Write. Druhý pomocný automat SENDER obsluhuje předávání dat načtených do fronty na sběrnici LAD. Využíván je rovněž u operace PCI Target Read a PCI Initiator Write. V rámci těchto operací zpracovává také čekací takty vložené ze strany PLX a je schopen tyto v případě potřeby sám iniciovat. Schéma rozdělení vstup/výstupních a interních řídicích signálů mezi všechny tři prvky centrálního systému je na obrázku 3.18.



Obrázek 3.18 Schéma zapojení centrálního řídicího systému

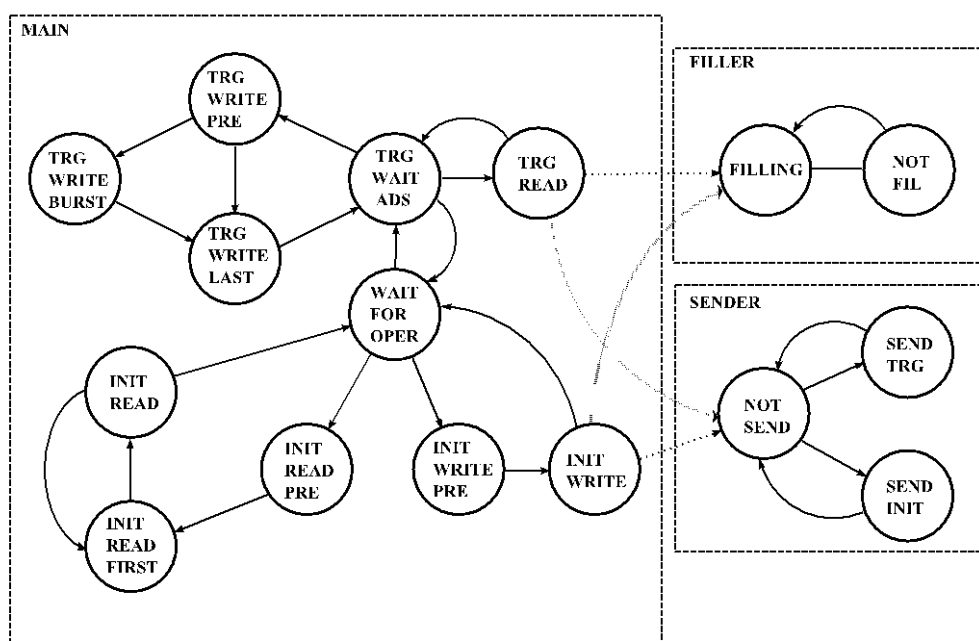
Pro kontrolu počtu načtených a odeslaných dat byla řídicí logice dána k dispozici 10b sčítačka (dále počítadlo odeslaných dat), paralelně plněná při inicializaci operací skrze rozšířené rozhraní signálem LEN. Odečítání počtu předaných dat provádí u operace PCI Initiator Read hlavní automat MAIN (sig. main\_dec\_cnt) a u operace PCI Initiator Write pomocný automat SENDER (sig. sender\_dec\_cnt).

**Hlavní konečný automat MAIN** – jedná se o synchronní automat s asynchronním resetem Mealyho typu, obsahuje 11 stavů:

- WAIT FOR OPER,
- TRG WAIT ADS,
- TRG WRITE PRE,
- TRG WRITE BURST,
- TRG WRIE LAST,
- TRG READ,
- INIT WRITE PRE,
- INIT WRITE,
- INIT READ PRE,
- INIT READ FIRST,
- INIT READ.

Obecný graf návaznosti stavů je zobrazen na obrázku 3.19. Počátečním stavem je WAIT\_FOR\_OPER, po shození signálu RESET se automat nachází právě v tomto stavu.

Pro operace v režimu PCI Target je vyhrazeno 5 navazujících stavů, z čehož čtyři jsou určeny pro operaci Write a jeden pro operaci Read. Při operaci Read dochází k zahájení činnosti obou



Obrázek 3.19 Graf přechodů konečných automatů MAIN, FILLER a SENDER

pomocných automatů. Pro operace v režimu PCI Initiator je vyhrazeno také 5 stavů, dva pro operaci Write a tři pro operaci Read. K aktivaci pomocných automatů dochází pouze při operaci Write.

**Pomocný konečný automat FILLER** – sestává ze dvou stavů, NOT FILL a FILLING. Jedná se o synchronní automat Mealyho typu s asynchronním resetem. Počátečním stavem je NOT FILL. Jeho činnost je zahájena signálem fill nastavovaným automatem MAIN v průběhu operací PCI Target Read, PCI Initiator Write – tedy u operací, při nichž dochází k přenosu dat z uživatelské komponenty do PLX za využití fronty FIFO.

**Pomocný konečný automat SENDER** – sestává ze tří stavů, NOT SEND, SEND INIT, SEND TRG. Pracuje na synchronním principu s asynchronním resetem, je Mealyho typu. Počátečním stavem je stav NOT SEND. Činnost automatu zahajuje hlavní automat MAIN signálem send. Je využíván u stejných operací jako automat FILLER, v případě operace PCI Target Read automat přechází do stavu SEND TRG, u operace PCI Initiator Write pak do stavu SEND INIT. U operace v režimu Initiator sám obsluhuje čítač odeslaných dat, u operace Target sleduje vystavení signálu k ukončení operace z PLX (BLAST). Uvedené dokončení operace (čítač roven nule, signál BLAST) sděluje automatu MAIN signálem sent.

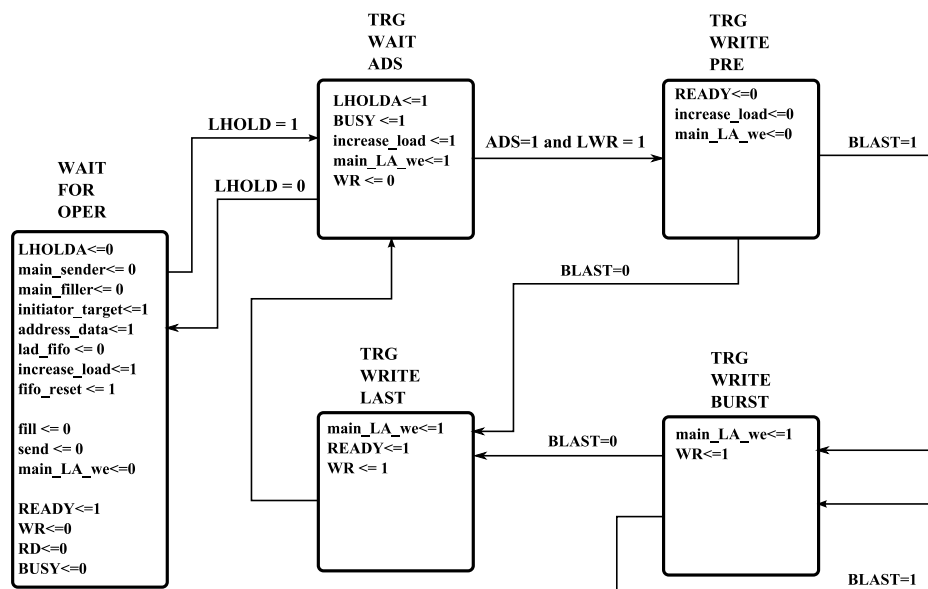
Jak již bylo naznačeno v průběhu textu rozšířená komponenta sestávající z výše popsaných funkčních celků umožňuje provádění **dvou druhů operací, každá ve dvou různých režimech**. Ve zbytku této kapitoly bude popsána činnost jednotlivých částí komponenty v průběhu každé z těchto operací se zaměřením na činnost centrálního řídicího systému.

**Operace Zápis** v režimu **PCI Target** (obr. 3.20) je realizována posloupností čtyř stavů v případě přenosu jednoho datového bloku a pěti stavů v případě dávkového přenosu. Konečný automat setrvává v počátečním stavu (WAIT\_FOR\_OPER) až do případného nastavení požadavku na přenos jádrem PLX na vodiči LHOLD. Na tuto odpovídá signálem LHOLDA a přechází do stavu očekávání adresy (TRG\_WAIT\_ADS). Současně nastavuje plnění registru LAREG ze sběrnice LAD (increase\_load) a povoluje jeho zápis (main\_LA\_we). Jakmile je vystavena platná adresa (ADS) a požadovaná operace je zápis (LWR) konečný automat potvrdí zahájení operace (READY) a registr LAREG uvede do režimu zvyšování adresy (TRG\_WRITE\_PRE). Registr DIREG je průběhu následující fáze operace plněn automaticky ze sběrnice LAD. Obsah registrů DIREG a LAREG je následně předáván uživatelské komponentě (WAIT\_DATA\_BURST), operace je indikována signálem WR. Data i adresa jsou uživatelské komponentě předávána se zpožděním jednoho taktu. Po obdržení posledního bloku dat (BLAST) automat operaci ukončí (TRG\_WRITE\_LAST) a vrací se do stavu čekání na adresu (TRG\_WAIT\_ADS). Jedná-li se o jednoduchý přenos, tzn. první datový blok je obdržen společně se signálem BLAST, automat přechází přímo do stavu WRITE\_DATA\_LAST. Úplný diagram průběhu operace zápis v režimu je na obázku 3.19.

V průběhu celé operace je rozšířená komponenta nastavena na režim Target (initiator\_target), sběrnice LAD je nastavena jako vstupní (lad\_fifo), řízení zápisu do registru

LAREG(main\_filler) a obsluha signálu READY (main\_sender) jsou svěřeny pouze automatu MAIN. Pomocné automaty SENDER a FILLER jsou v průběhu operace neaktivní.

Dojde-li v průběhu operace ke vložení čekacích taktů ze strany PLX (WAIT), komponenta přeruší zápis do uživatelské komponenty (WR), pozastaví zápis do registru LAREG a vyčkává opětovnému pokračování transakce.



Obrázek 3.20 Diagram stavů konečného automatu MAIN u operace PCI Target Write

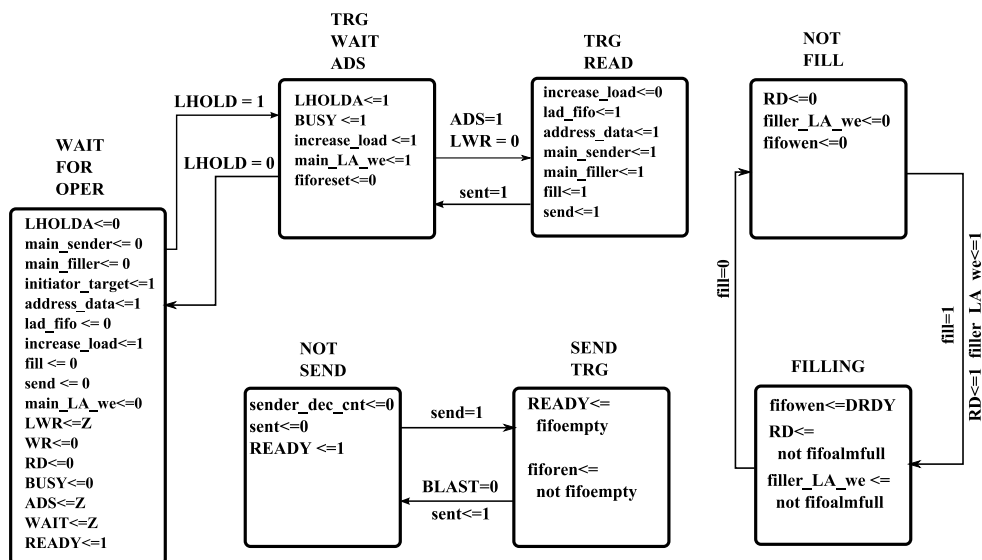
**Operace Čtení** v režimu **PCI Target** (obr. 3.21) se skládá ze tří stavů a lze ji rozdělit do dvou fází

- iniciace, načtení a předání adresy (PLX -> uživatelská komponenta)
- načítání a předávání dat (uživatelská komponenta -> PLX)

V první fázi pracuje pouze konečný automat MAIN, který setrvává v počátečním stavu (WAIT\_FOR\_OPER) až do případného nastavení požadavku na přenos jádrem PLX na vodiči LHOLD. Na tuto odpovídá signálem LHOLDA a přechází do stavu očekávání adresy (TRG\_WAIT\_ADS). Současně nastavuje plnění registru LAREG ze sběrnice LAD (increase\_load) a povoluje jeho zápis (main\_LA\_we). Jakmile je vystavena platná adresa (ADS) a požadovaná operace je čtení (LWR) operace přechází do druhé fáze (TRG\_READ). Zde dochází k aktivaci obou pomocných automatů (send, fill), obsluha signálu READY je předána automatu SEND (main\_sender), povolení zápisu do registru LAREG řídí automat FILLER (main\_filler), současně je provedena změna režimu tohoto registru na zvyšování adresy, sběrnice LAD je nastavena jako výstupní (lad\_fifo) a je plněna výstupní hodnotou fronty (address\_data). Pomocný automat FILLER kontinuálně plní frontu daty načtenými z uživatelské komponenty. Tuto činnost přerušuje pouze, pokud je fronta plná (fifoalmlfull). Druhý automat SENER vyčkává a jakmile jsou ve frontě přítomna načtená data (fifoempty) tyto postupně

předává na sběrnici LAD (fiforen). Takto pokračuje až do obdržení pokynu k ukončení operace (BLAST). Tento stav signalizuje hlavnímu automatu MAIN (sent), který ukončí druhou fázi a tak i celou operaci.

V době, kdy nejsou ve frontě data k dispozici, vkládá automat SENDER čekací stavy (READY). V případě, že dojde k vložení čekacích stavů ze strany PLX (WAIT), SENDER pouze pozastaví čtení z fronty (fiforen).

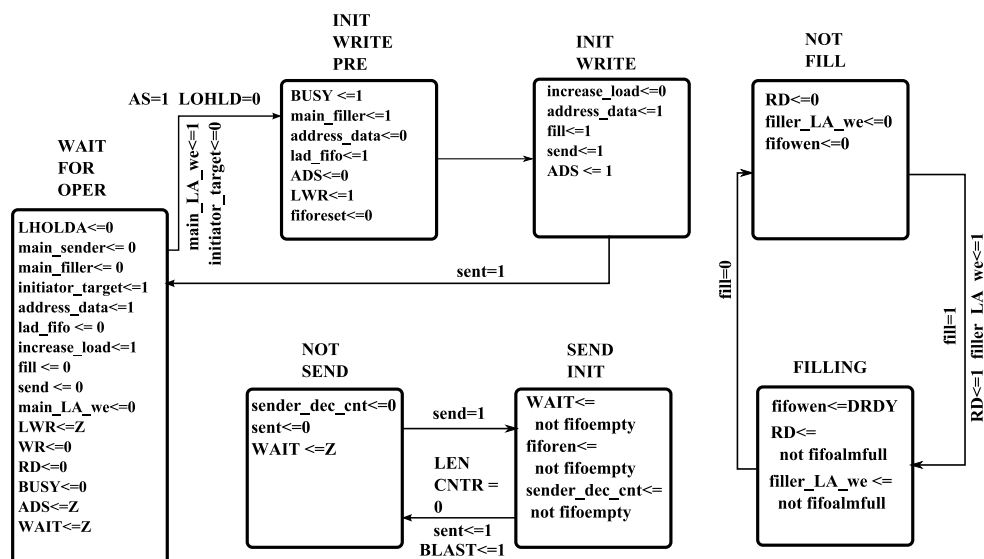


Obrázek 3.21 Diagram stavů konečných automatů u operace PCI Target Read

**Operace Zápis** v režimu **PCI Initiator** (obr. 3.22) je složena ze tří stavů konečného automatu MAIN. Tento setrvává v počátečním stavu (WAIT\_FOR\_OPER) až do požadavku na přenos dat ze strany uživatelské komponenty na vodiči AS. Je-li tento nastaven, dojde k uložení všech parametrů přenosu - LA do LAREG, MA do MAREG, LEN do počítadla odeslaných dat (LEN\_CNTR). Automat MAIN zahájení operace potvrzuje signálem BUSY. V následujícím hodinovém taktu je nastaven druh operace jako zápis (LW) a cílová adresa odeslána skrze PLX rozhraní tak, že sběrnice LAD je nastavena jako výstupní (lad\_fifo) a na její vstup je nasměrován obsah registru MAREG (address\_data). Po odeslání adresy dojde k aktivaci obou pomocných automatů (send, fill), současně je předáno řízení povolení zápisu do registru LAREG automatu FILLER (main\_filler), režim tohoto registru je změněn na zvyšování adresy (increase\_load) a sběrnice LAD je plněna výstupní hodnotou fronty (address\_data). Pomocný automat FILLER kontinuálně plní frontu daty načtenými z uživatelské komponenty. Tuto činnost přerušuje pouze, pokud je fronta plná (fifoeempty). Druhý automat SENER vyčkává a jakmile jsou ve frontě přítomna načtená data (fifoeempty) tyto postupně předává na sběrnici LAD (fiforen) za současného snižování počítadla odeslaných dat (sender\_dec\_cnt). Takto pokračuje do doby, kdy je počítadlo rovno nule. Tento stav signalizuje hlavnímu automatu MAIN (sent), který ukončí činnost automatu FILLER (fill) a následně i celou operaci. V době,

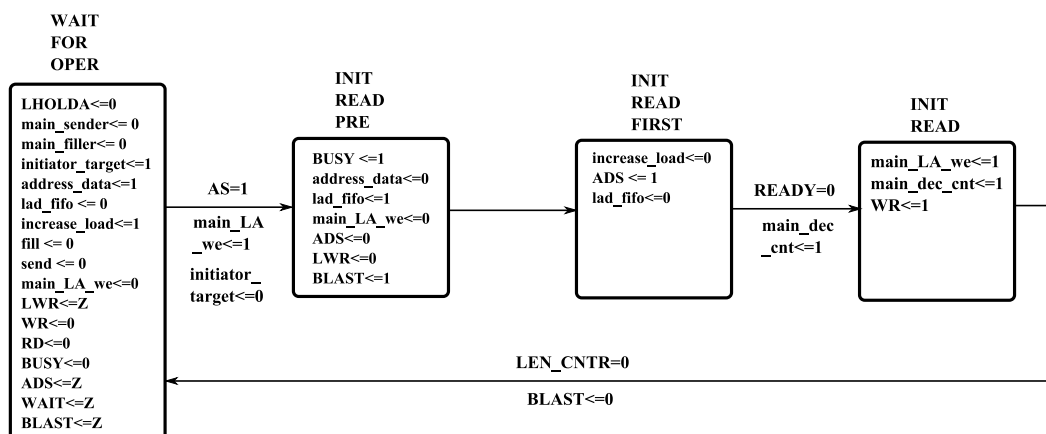


kdy nejsou ve frontě data ještě nebo již k dispozici, vkládá automat SENDER čekací stavy (WAIT). Dojde-li k vložení čekacích stavů také ze strany PLX (READY), SENDER pouze pozastaví čtení z fronty (fiforen).



Obrázek 3.22 Diagram stavů konečných automatů u operace PCI Initiator Write

**Operace Čtení** v režimu **PCI Initiator** (obr. 3.22) je provedena průchodem 4 stavů hlavního automatu MAIN. Obdobně jako u předchozí operace automat setrvává v počátečním stavu



Obrázek 3.23 Diagram stavů konečných automatů u operace PCI Initiator Read

(WAIT\_FOR\_OPER) až do požadavku na přenos dat ze strany uživatelské komponenty na vodiči AS. Je-li tento nastaven, dojde k uložení všech parametrů přenosu - LA do LAREG, MA do MAREG, LEN do počítadla odeslaných dat (LEN\_CNTR). Automat MAIN zahájení operace potvrzuje signálem BUSY. V následujícím hodinovém taktu je nastaven druh operace jako čtení (LW) a zdrojová adresa odeslána skrze rozhraní PLX tak, že sběrnice LAD je nastavena jako výstupní (lad\_fifo) a na její vstup je nasměrován obsah registru MAREG (address\_data). Po odeslání adresy dojde k opětovné rekonfiguraci sběrnice LAD na vstupní (lad\_fifo) a je vyčkáváno načtení požadovaných dat. Po jejich příchodu (READY) jsou tyto ukládány do registru DIREG a po jednom taktu pak společně s platnou adresou z registru LAREG předány uživatelské komponentě včetně indikace operace zápis (WR). Současně je snižována hodnota počítadla odeslaných dat (main\_dec\_cnt). Jakmile je hodnota počítadla rovna nule, čtení dat skrze PLX je ukončeno (BLAST). Dojde-li v průběhu operace k vložení čekacích stavů ze strany PLX (READY), automat MAIN pouze dočasně pozastaví předávání dat uživatelské komponentě a přeruší postupné zvyšování adresy v registru LAD (WR).

## 4 Simulace a syntéza

Ověření funkčnosti hardwarových zařízení probíhá v sérii několika na sebe navazujících kroků. Každý krok je zaměřen na zkoumání z jiného úhlu pohledu a tyto na sebe vzájemně navazují. U navržených komponent jsou s ohledem na vstupní podmínky sledovány nejen jejich vnější projevy, nýbrž jsou vyhodnocovány i děje probíhající uvnitř. Pro ověření absolutní funkčnosti zařízení by bylo ideální toto otestovat na množině všech možných kombinací vstupních podmínek. To však vzhledem k obrovskému počtu prvků této množiny není reálně možné. V rámci testování je tedy nutné stanovit určitou podmnožinu těchto podmínek, která zařízení prověří alespoň, co se týče typických standardních a nestandardních situací.

Kapitola popisuje metody testování obou navržených komponent včetně prezentace základních výsledků. Série testů byla rozdělena do dvou kroků – simulace navržené komponenty před syntézou a provedení syntézy zařízení pro konkrétní FPGA.

### 4.1 Testovací prostředí

V rámci testování bylo využito nástrojů společnosti XILINX Inc. obsažených ve vývojovém balíčku Xilinx ISE Webpack, v. 13.1, volně stažitelného z internetové stránky [www.xilinx.com](http://www.xilinx.com). Balíček je pro nekomerční využití bez poplatku.

#### Využité nástroje:

- Simulace: **ISIM release version 13.1 application version: 0.40d**
- Syntéza: **Xilinx XST v. 13.1**

### 4.2 Základní komponenta

Základní komponenta pracuje dle své specifikace pouze na dvou rozhraních, a to rozhraním s PLX a základním rozhraním s uživatelskou komponentou. Provádí dvě operace čtení a zápis v režimu PCI Target. S přihlédnutím k uvedenému byla navržena série testů sestávající ze 3 kroků:

- (1) Behaviorální simulace základní komponenty:
  - operace zápis v PCI Target – jednoduchá i dávková,
  - operace čtení v PCI Target – jednoduchá i dávková.
- (2) Syntéza základní komponenty,

(3) Behaviorální simulace zařízení v „toplevel“ entitě s uživatelskou komponentou BlockRam (512x32b):

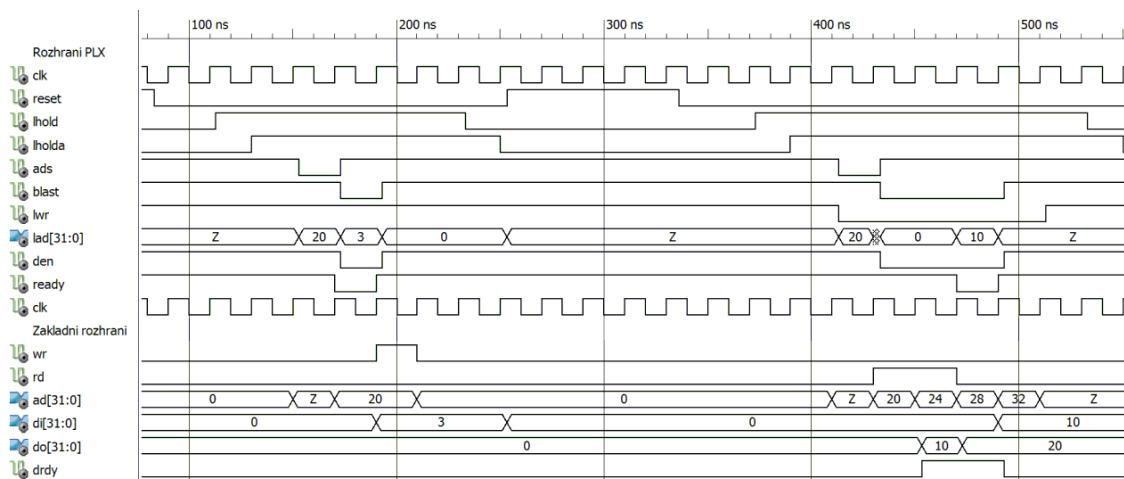
- operace zápis v PCI Target – jednoduchá i dávková,
- operace zápis v PCI Target – jednoduchá i dávková.

(1) **Behaviorální simulace samotného zařízení** (obr. 4.1) – základní komponenta je vložena do testovacího návrhu (tzv. testbench) obsahujícího simulační procedury. Každá operace je obsažena v jedné ze simulačních procedur. Procedury jsou navrženy tak, aby v příslušnou dobu, a to i s ohledem na výstupní signály základní komponenty, generovaly signály odpovídající protokolům užitým na daných rozhraních. V rámci vyhodnocení testu jsou pak zkoumány reakce základní komponenty.



Obr. 4.1 Schéma zapojení při behaviorální simulaci samotné základní komponenty

Výsledky simulace u jednoduchých operací jsou na následujícím obrázku 4.2, výsledky dávkových operací se nacházejí v příloze D.



Obrázek 4.2. Průběh operace zápis a čtení 1 bloku dat

(2) **Syntéza základní komponenty** provedla překlad navrženého zařízení s ohledem na jeho optimalizaci a následné umístění do programovatelného hradlového pole. Výsledkem syntézy je mimo jiné informace o počtu hardwarových prostředků FPGA spotřebovaných komponentou včetně důležitých časových informací o tocích signálu uvnitř komponenty (např. kritická cesta).

Syntéza základní komponenty na FPGA Spartan 3 typ X3S200 nevykázala žádné varování ani chyby.

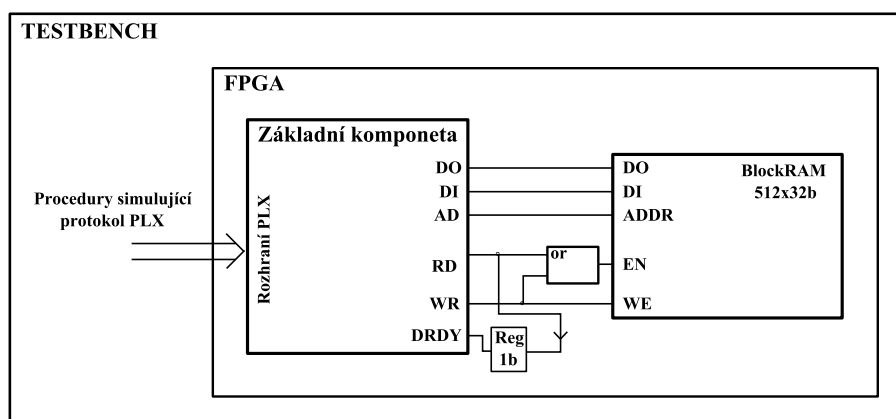
Nároky na HW prostředky FPGA Spartan 3 typ X3S200:

- Slices: 42 z 1920 2%
- Slice Flip Flops: 39 z 3840 1%
- 4 input LUT: 79 z 3840 2%
- GCLK: 1 z 8 12%

Časové nároky:

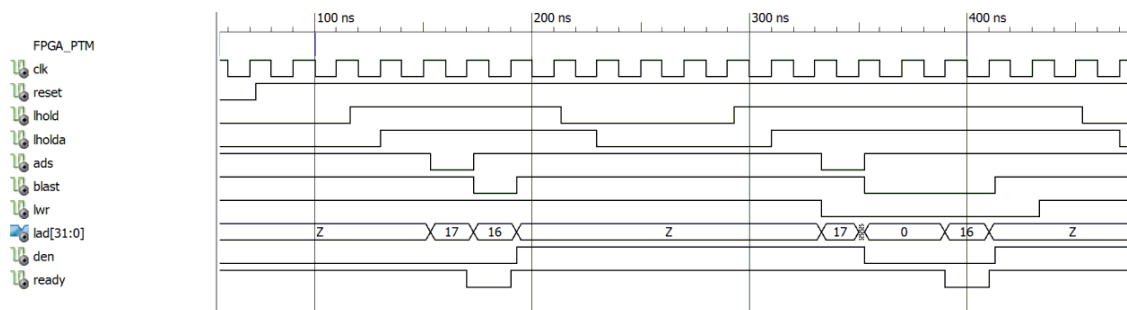
- Minimální perioda: 7.242ns (**Maximální frekvence: 138.083MHz**)
- Minimální doba změny vstupu před nástupnou hranou hodin: 4.628ns
- Maximální době změny výstupu po nástupné hraně hodin: 10.267ns
- Doba nejdelší kombinační cesty: 8.933ns

(3) **Behaviorální simulace v „toplevel“ entitě FPGA s uživatelskou komp. BlockRam 512x32** (obr. 4.3) – cílem testu je prověření funkce s konkrétní uživatelskou komponentou. Touto byl vybrán referenční návrh paměti BlockRam od společnosti XILINX[13]. Paměť je jednoportová, velikost 512x32b. Pro účely testu bylo upraveno propojení těchto komponent tak, aby si vzájemně odpovídali jejich protokoly. Vloženým registrem je simulováno zpoždění jednoho aktu mezi požadavkem na čtení (RD) a vystavením konkrétních dat (DRDY).



Obr. 4.3 Schéma zapojení při behaviorální simulaci základní komponenty s BlockRAM

Testování proběhlo pouze pomocí procedur simulujících protokol na straně PLX. Výsledky simulace jednoduché operace zápis a čtení jsou na obrázku 4.4.



Obrázek 4.4. Průběh operace zápis a čtení 1 bloku dat se zapojenou BlockRAM

### 4.3 Rozšířená komponenta

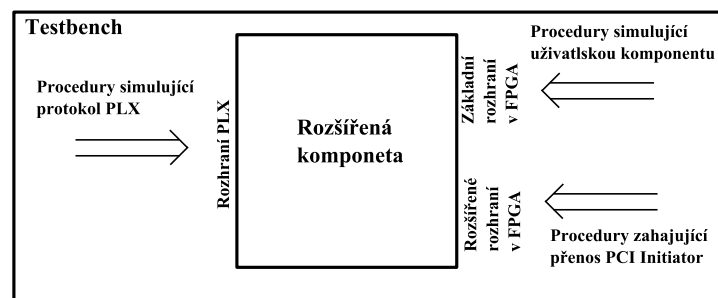
Rozšířená komponenta pracuje dle své specifikace pouze na všech třech rozhraních. Provádí operace čtení a zápis v režimech PCI Target a PCI Initiator. S přihlédnutím k uvedenému byla navržena série testů sestávající ze 2 kroků:

(1) Behaviorální simulace rozšířené komponenty:

- operace zápis v PCI Target – jednoduchá i dávková,
- operace čtení v PCI Target – jednoduchá i dávková,
- operace zápis v PCI Initiator – jednoduchá i dávková,
- operace čtení v PCI Initiator – jednoduchá i dávková,

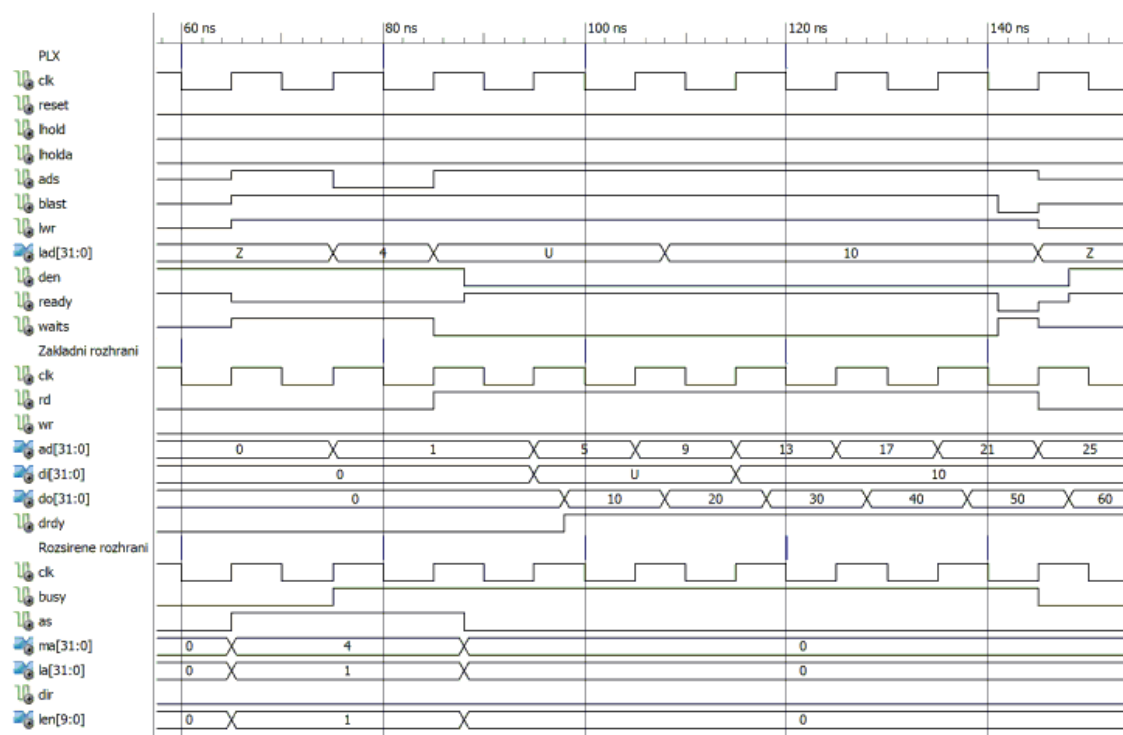
(2) Syntéza rozšířené komponenty,

**(1) Behaviorální simulace rozšířené komponenty** (obr. 4.5) - tato je vložena do testovacího návrhu (tzv. testbench) obsahujícího simulační procedury. Každá operace je obsažena v jedné ze simulačních procedur. Procedury jsou navrženy tak, aby v příslušnou dobu, a to i s ohledem na výstupní signály rozšíření komponenty, generovaly signály odpovídající protokolům užitým na daných rozhraních. Současně jsou v průběhu operací vkládány čekací stavy. V rámci vyhodnocení testu jsou pak zkoumány reakce rozšířené komponenty.



Obr. 4.5 Schéma zapojení při behaviorální simulaci rozšířené komponenty

Simulací bylo zjištěno, že rozšířená komponenta na svých výstupech generuje signály odpovídající navrženým protokolům i protokolu s PLX. Příklad výsledku operace zápis v režimu Initiator je na obr. 4.6, výsledky všech operací se nacházejí v příloze G.



Obrázek 4.6. Průběh operace zápis v režimu PCI Initiator - jednoduchý

(2) **Syntéza rozšířené komponenty** provedla překlad navrženého zařízení s ohledem na jeho optimalizaci a následné umístění do programovatelného hradlového pole. Výsledkem syntézy je mimo jiné informace o počtu hardwarových prostředků FPGA spotřebovaných komponentou včetně důležitých časových informací o tocích signálu uvnitř komponenty (např. kritická cesta).

Syntéza rozšířené komponenty na FPGA Spartan 3 typ X3S200 nevykázala žádné varování ani chyby.

Nároky na HW prostředky FPGA Spartan 3 typ X3S200:

- Slices: 168 z 1920 8%
- Slice Flip Flops: 116 z 3840 3%
- 4 input LUT: 320 z 3840 8%
- GCLK: 1 z 8 12%

Časové nároky:

- Minimální perioda: 12,196ns (**Maximální frekvence: 81.966MHz**)
- Minimální doba změny vstupu před nástupnou hranou hodin: 8.934ns
- Maximální době změny výstupu po nástupné hraně hodin: 16.734ns
- Doba nejdelší kombinační cesty: 13.417ns



## 5 Závěr

Cílem této práce bylo navrhnout architekturu a provést implementaci zařízení do programovatelného hradlového pole FPGA, které bude schopno zprostředkovat komunikaci se sběrnici PCI reprezentovanou vestavěným jádrem PLX, přičemž na straně aplikací umístěných v FPGA výrazně zjednoduší komunikační protokol za současného zachování pokročilých funkcí sběrnice PCI (DMA přenosy).

Za tímto účelem bylo provedeno seznámení se specifikacemi použité akcelerační karty COMBO6-PTM a vestavěného jádra pro komunikaci s PCI sběrnici PLX. Současně byl prostudován komunikační protokol samotné sběrnice PCI včetně problematiky týkající se přímého přístupu do paměti. Načerpané znalosti byly použity při tvorbě teoretického úvodu.

Na základě získaných vědomostí bylo rozhodnuto o vytvoření dvou komponent – základní verze šetřící hardwarové prostředky programovatelného hradlového pole bez DMA přenosů, a její rozšířené verze umožňující přenosy DMA. Návrh architektury byl proveden tak, aby mohla být provedena implementace a následná syntéza do programovatelného hradlového pole Spartan 3 typ X3S200. Současně s návrhem architektury byl vytvořen také zjednodušený komunikační protokol určený pro komunikaci s dalšími zařízeními uvnitř cílového FPGA.

Obě komponenty byly implementovány v jazyce VHDL. Následně prošly sérií simulací, které ověřili jejich funkčnost. Provedenou syntézou byly zjištěny konkrétní nároky na HW prostředky FPGA včetně důležitých časových údajů především pak maximálního kmitočtu, na kterém jsou komponenty schopny pracovat.

Prototyp základní komponenty byl úspěšně realizován na kartě COMBO6-PTM. Testování rozšířené komponenty nebylo v hardware provedeno, neboť komponenta PLX pracuje v režimu Initiator pouze po specifickém nastavení jeho interních registrů. Tyto registry mohou být nastavovány skrze sériově plněnou paměť EEPROM případně vhodnou aplikací přímo ze sběrnice PCI. Obsluha interních registrů jádra PLX je však již nad rámec této práce.

V rámci dalšího pokračování projektu je možné obě zařízení upravit tak, aby byly schopny pracovat s rozlišnou šířkou datového přenosu na straně PLX. Současně by mohla být vytvořena aplikace nebo chybějící zařízení určené k nastavování interních registrů PLX, na základě kterého by pak byla realizována i rozšířená komponenta na kartě COMBO6-PTM. Přestože rozšířená komponenta splňuje požadavek na práci při kmitočtu 50Mhz, bylo by vhodné ji v rámci optimalizace analyzovat a následně upravit tak, aby se snížila délka nejdelší kombinační cesty. Komponenta by tak mohla pracovat na vyšším kmitočtu s cílem dosáhnout hodnoty 100Mhz.

# Literatura

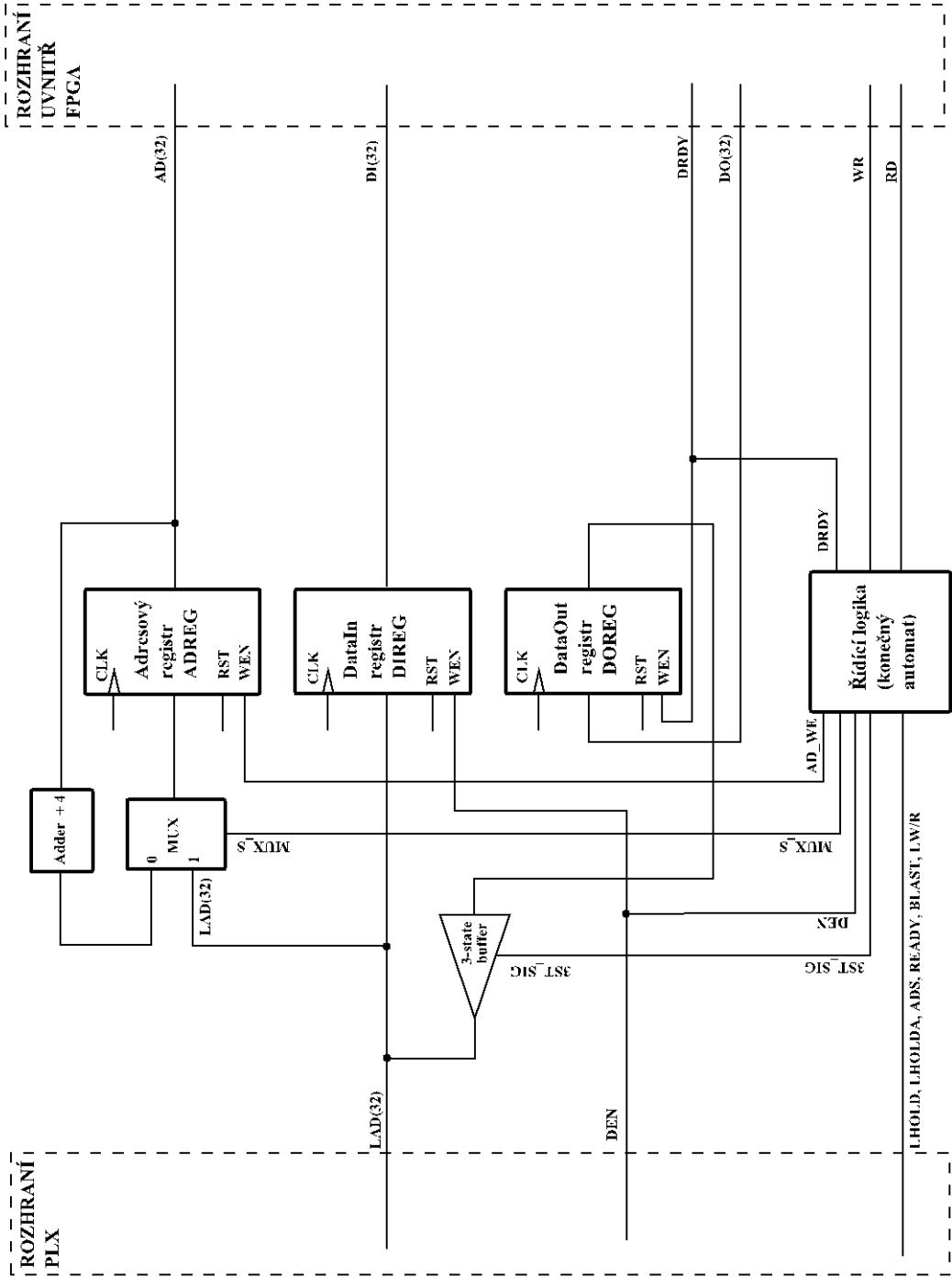
- [1] Liberouter project[online]. Dostupný 10.1.2011 na URL: < <http://www.liberouter.org>>
- [2] LIPSON, A., HANZELHURST, L., HANZELHURST, S.: DNA Pattern Matching using FPGAs[online]. Dostupný 2.1.2011 na URL: <<http://www.prasa.up.ac.za/GeneralPatternRecognitionII/PDF/G8ALipson.pdf>>
- [3] MAHMOUD, B., BEDOUI, M.H., RAYCHEV, R., ESSABBAH, H. (2005): Nuclear medical image treatment system based on FPGA in real time[online]. Dostupný 2.1.2011 na URL: <<http://www.waset.org/journals/ijice/v1/v1-2-12.pdf>>
- [4] NOVOTNÝ, J., FUČÍK, O., BARDAS, R. (2003): Schematic of COMBO-PTM, Technical report[online]. Dostupný 10.11.2010 na URL: <<http://www.cesnet.cz/doc/techzpravy/2003/comboptmschematic/comboptmschematic.pdf>>
- [5] COMBO6-PTM[online]. Dostupný 10.1.2011 na URL: <[http://www.liberouter.org/card\\_comboptm.php](http://www.liberouter.org/card_comboptm.php)>
- [6] TIŠNOVSKÝ, P. (2008): Sběrnice VESA Local Bus a universální sběrnice PCI[online]. Dostupný 4.1.2011 na URL < <http://www.root.cz/clanky/sbernice-vesa-local-bus-a-universalni-sbernice-pci/>>
- [7] PCI(sběrnice)[online]. Dostupný 4.1.2011 na URL: <[http://cs.wikipedia.org/wiki/PCI\\_\(sb%C4%9Brnice\)](http://cs.wikipedia.org/wiki/PCI_(sb%C4%9Brnice))>
- [8] How the PCI Bus Works[online]. Dostupný 5.1.2011 na URL: <[http://www.tech-pro.net/intro\\_pci.html](http://www.tech-pro.net/intro_pci.html)>
- [9] RANJITH(2009): Introduction to PCI protocol[online]. Dostupný 2.1.2011 na URL <<http://electrofriends.com/articles/computer-science/protocol/introduction-to-pci-protocol/4/>>
- [10] DMA[online]. Dostupný 2.1.2011 na URL: <<http://cs.wikipedia.org/wiki/DMA>>
- [11] HENNESSY, J. L., PATTERSON, D. A.: Computer Architecture A Quantitative Approach. 3. Vydání. Morgan Kaufman Publishers, 2003. 883s. ISBN 1-55860-596-7.
- [12] PLX TECHNOLOGY(2000): PCI 9054 Data Book[online]. Dostupný 3.1.2011 na URL: <<http://attila.kinali.ch/ogp/datasheets/unused/plx/9054db-21.pdf>>
- [13] XILINX, Inc.(2005): Using Block RAM in Spartan-3 Generation FPGAs[online]. Dostupný 10.4.2011 na URL: <[http://www.xilinx.com/support/documentation/application\\_notes/xapp463.pdf](http://www.xilinx.com/support/documentation/application_notes/xapp463.pdf)>

# Příloha A - Obsah příloženého CD

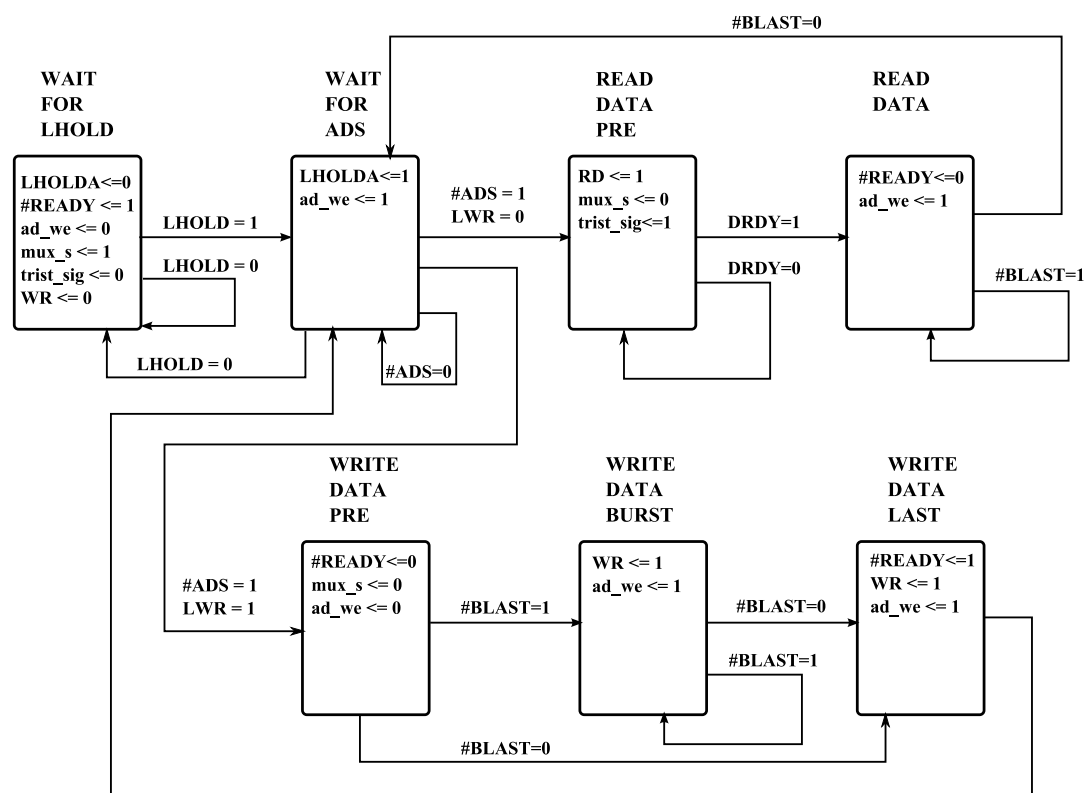
Adresářová struktura příloženého CD je následující:

- v kořenovém adresáři soubor *readme.txt* obsahuje uživatelskou příručku - návod ke spuštění a simulaci všech implementovaných komponent včetně popisu postupu při překladu zdrojového kódu
- adresář *FIFO\_mod* – projekt s implementovanou komponentou fronta včetně testovacích procedur
- adresář *Zakladni\_komponenta* – projekt s implementovanou základní komponentou zapojenou samostatně včetně testovacích procedur
- adresář *FPGA\_zakladni\_blockram* – projekt s implementovanou základní komponentou v „top level“ entitě FPGA Spartan 3 zapojenou s BlockRAM 512 x 32b, entita FPGA převzata z předmětu Návrh adaptérů a vestavěných systému
- adresář *Zakladni\_HW* – základní komponenta v entitě FPGA s BlockRAM včetně procedur k překladu, syntéze a vložení do HW, převzata z předmětu Návrh adaptérů a vestavěných systému a následně upraveno pro konkrétní úlohu
- adresář *Rozsirena\_komponenta* – projekt s implementovanou rozšířenou komponentou zapojenou samostatně včetně testovacích procedur
- adresář *FPGA\_rozsirena\_blockram* – projekt s implementovanou rozšířenou komponentou zapojenou v „top level“ entitě FPGA Sparta 3 zapojenou s BlockRAM 512x32b, entita FPGA převzata z předmětu Návrh adaptérů a vestavěných systému
- adresář *thesis* – text diplomové práce v elektronické podobě

# Příloha B - Schéma základní komponenty

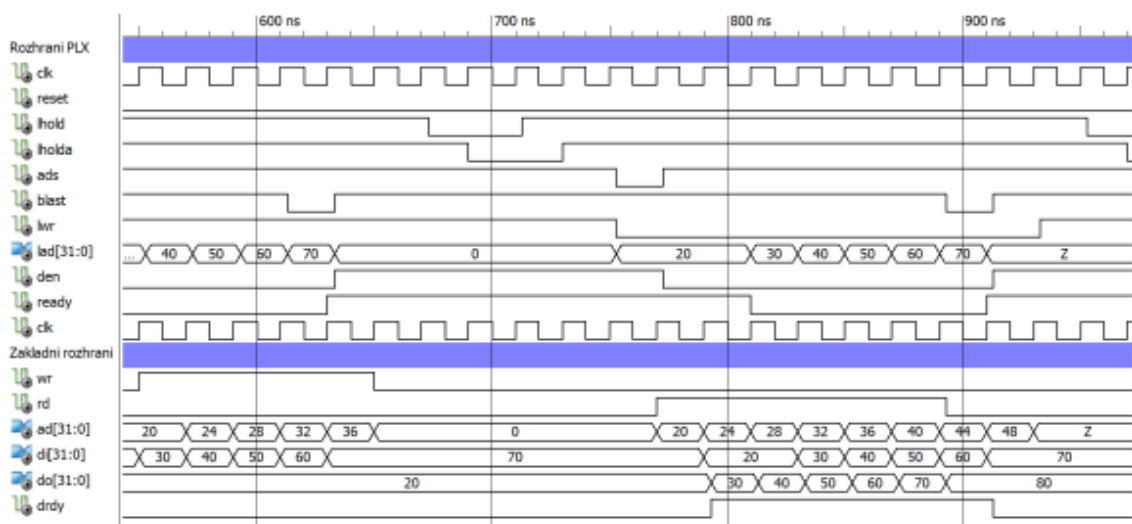
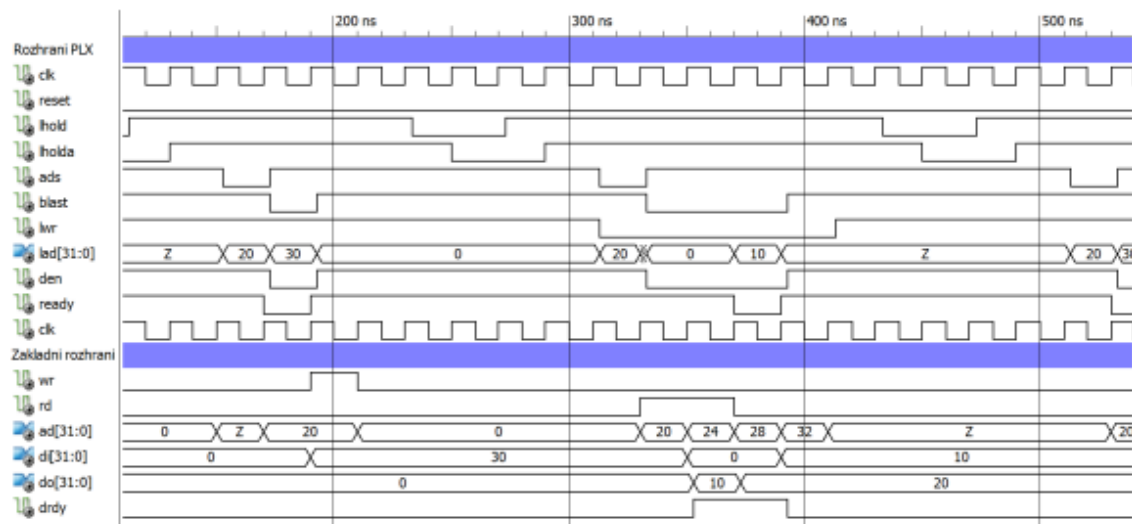


## Příloha C - Schéma konečného automatu základní komponenty

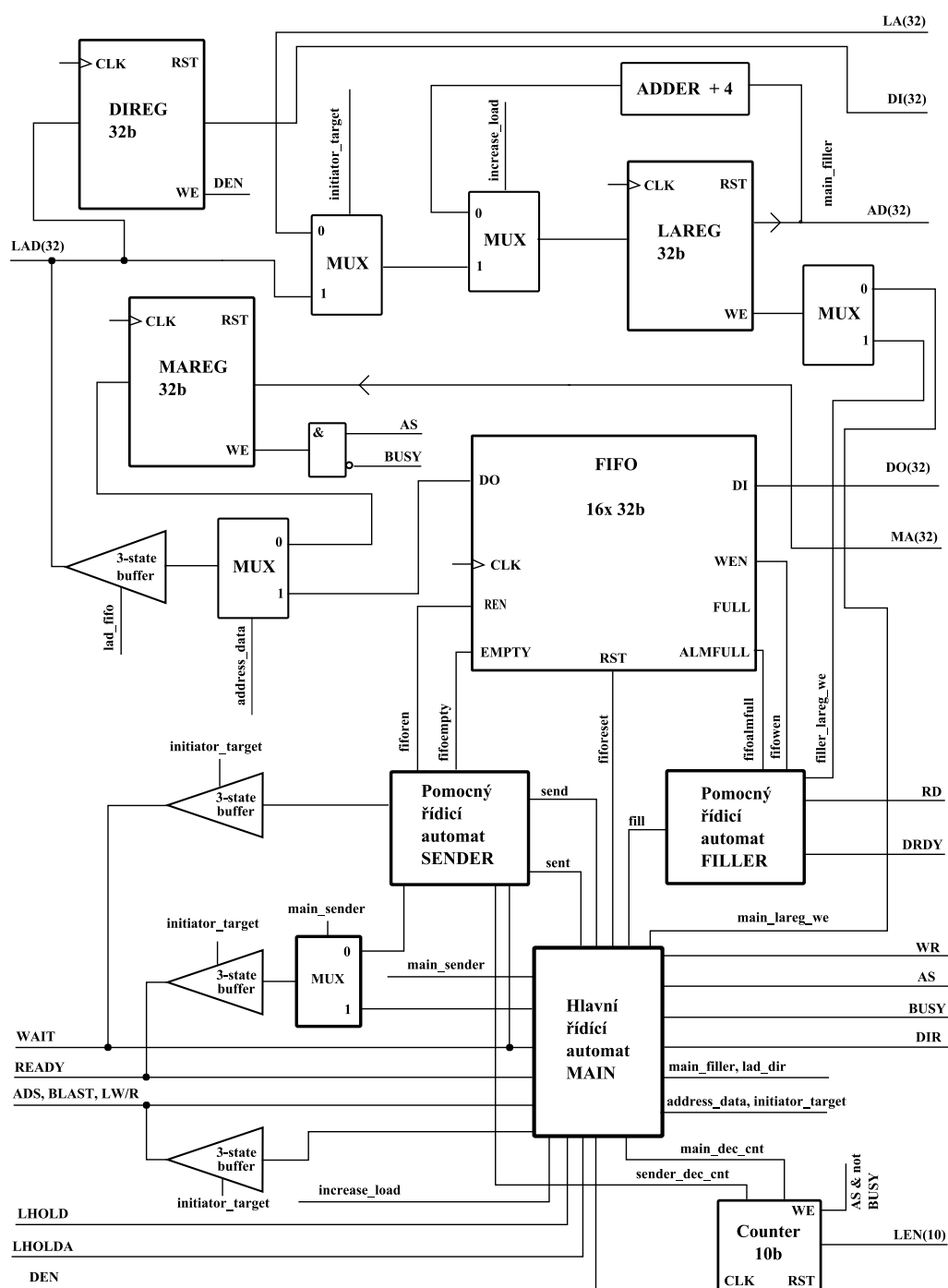


# 

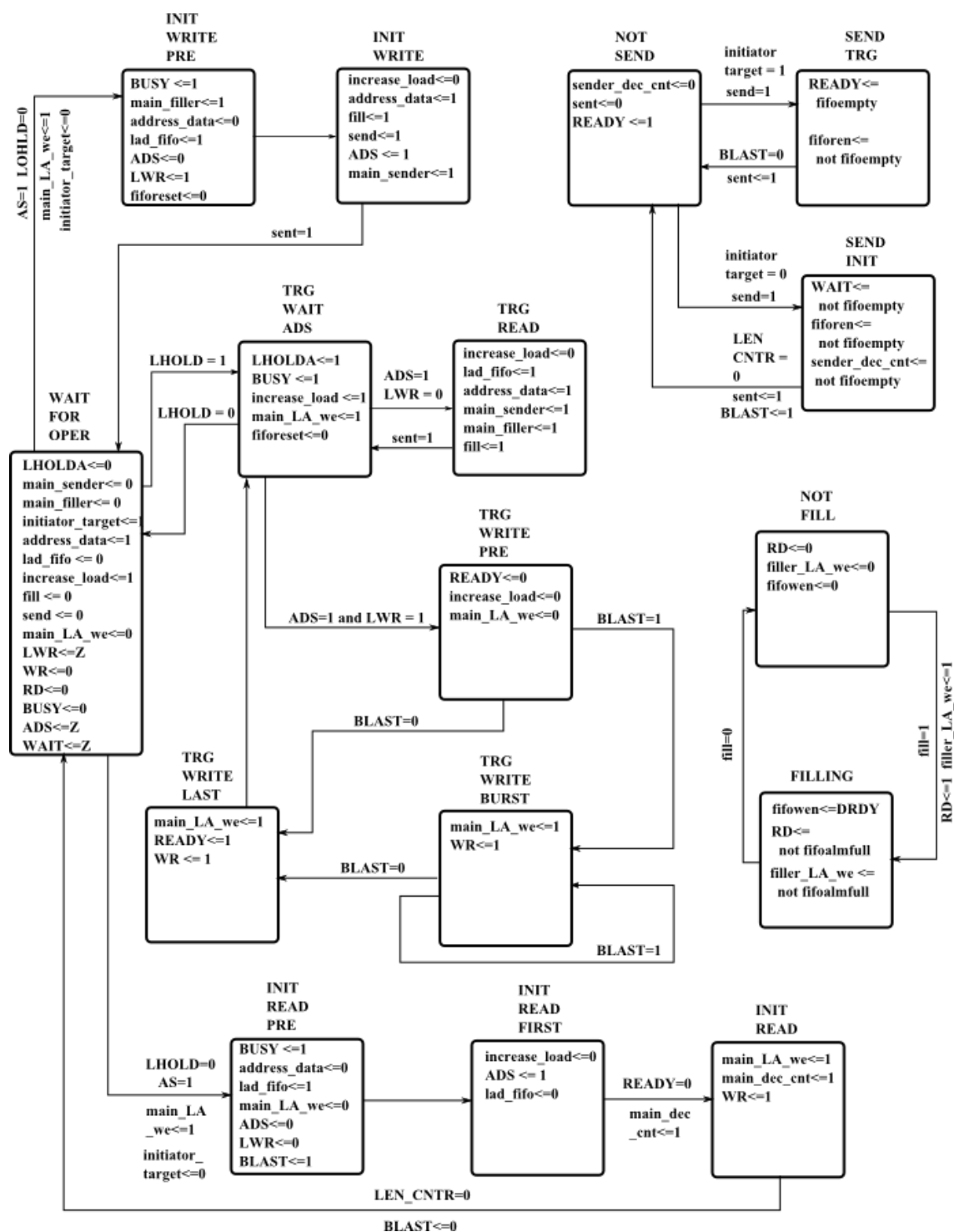
Čas. průběh operací 1x zápis, 1x čtení, 5x zápis a 5x čtení. Frekvence hod. signálu 50 MHz.



# 



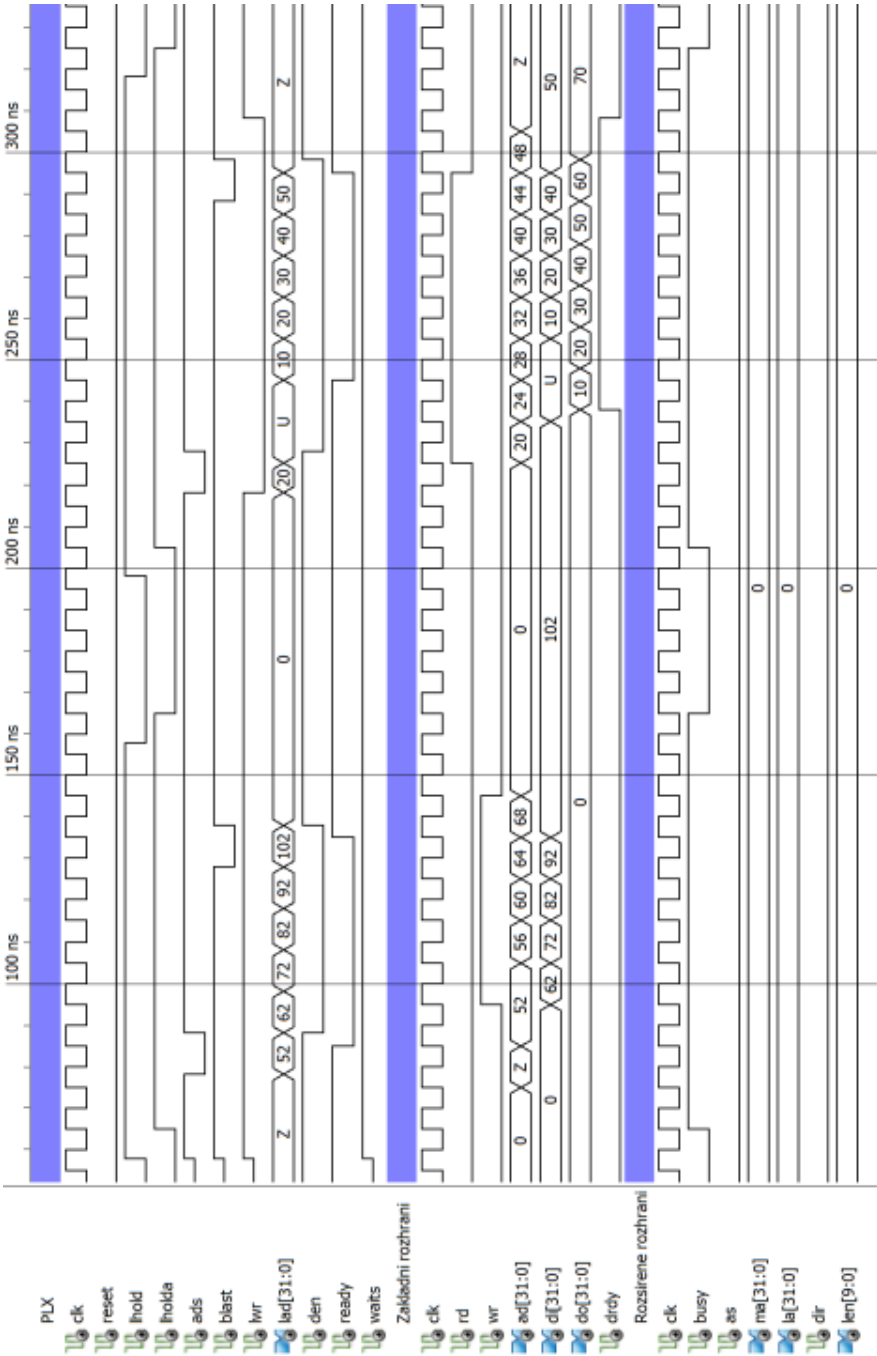
## Příloha F - Schéma konečného automatu rozšířené komponenty





# Příloha G - Časové diagramy operací rozšířené komponenty

Čas. průběh operací v režimu Target – 5x zápis a 5x čtení. Frekvence hod. signálu 50 Mhz.



Čas. průběh operací v režimu Initiator – 5x zápis a 5x čtení. Frekvence hod. signálu 50 Mhz.

